

Práctica Git

Puntos totales 12/13

En <https://github.com/kasappeal/Git-GKC/raw/main/Pr%C3%A1ctica.pdf> puedes encontrar el enunciado de la práctica. Consta de dos ejercicios, el primero obligatorio, el segundo es opcional. Sigue cada uno de los pasos que se indica en el ejercicio y apunta los comandos que vas utilizando en cada paso, ya que es lo que se preguntará en el test. El test evaluará el primer ejercicio. Puede haber más de una respuesta correcta para cada pregunta.

0 de 0 puntos

correo electrónico *

dianamtlopez@hotmail.com

Nombre *

DIANA MARIA

Apellidos *

TORO LOPEZ

Pregunta nº 1

1 de 1 puntos

¿Qué comando utilizaste en el paso 11? *

1/1

- ☐ git restore <archivo>
- ☒ git reset --hard HEAD~ 1 para descartar los cambios realizados
- ☐ git reset HEAD~1 y luego git restore <archivo>
- ☐ git accio

Pregunta nº 2

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 12? *

1/1

- ☐ Se podría realizar el merge de nuevo con el comando git merge
- ☐ git add <archivo> u luego git restore <archivo>
- ☒ git reflog y luego reset --hard HASH_COMMIT_PASO_10 para recuperar los cambios realizados
- ☐ git alohomora

Pregunta nº 3

1 de 1 puntos

El merge del paso 13, ¿Causó algún conflicto? *

1/1

- ☐ Sí, siempre que se hace un merge hay conflictos
- ☐ Sí, tocamos el mismo archivo en la misma línea en dos ramas diferentes
- ☒ No, porque no se ha realizado ningún merge ya que los commits de master ya están contenidos en la rama styled
- ☐ git expelliarmus

Pregunta nº 4

1 de 1 puntos

El merge del paso 19, ¿Causó algún conflicto?

*

1/1

- ☐ No, porque en master no hemos modificado nada
- ☒ Sí, tocamos el mismo archivo en la misma línea en dos ramas diferentes
- ☐ No, porque no se ha realizado ningún merge ya que los commits ya están contenidos
- ☐ git levicorpus

Pregunta nº 5

0 de 1 puntos

El merge del paso 21, ¿Causó algún conflicto?

*

0/1

- ☐ No, porque en master no hemos modificado nada
- ☐ Sí, tocamos el mismo archivo en la misma línea en dos ramas diferentes
- ☒ No, porque no se ha realizado ningún merge ya que los commits ya están contenidos
- ☐ git lumos

Respuesta correcta

- ☒ No, porque en master no hemos modificado nada

Pregunta nº 6

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 25? *

1/1

- ☐ git graph
- ☒ git log --graph
- ☐ git, arfavó de pintar las bolitas y las flechitas
- ☐ git wingardfium --leviosa

Pregunta nº 7

1 de 1 puntos

El merge del paso 26, ¿Podría ser fast forward? *

1/1

- ☒ Sí, porque title sólo está un commit por delante de master y forman una lista
- ☐ Sí, si usamos git merge --no-ff title
- ☐ No, git se ve forzado a crear un commit nuevo
- ☐ git imperius

Pregunta nº 8

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 27? *

1/1

- ☒ git reset HEAD~1
- ☐ git reset HEAD@{1}
- ☐ git restore <archivo>
- ☐ git patronus

Pregunta nº 9

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 28? *

1/1

- ☐ git checkout -- <archivo>
- ☐ git log
- ☒ git restore <archivo>
- ☐ git avada --kedavra

Pregunta nº 10

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 29? *

1/1

- ☐ git checkout title
- ☐ git restore title
- ☒ git branch -D title
- ☐ git sectumsempra

Pregunta nº 11

1 de 1 puntos

¿Qué comando o comandos utilizaste en el paso 30? *

1/1

- ☒ git reflog y luego git reset --hard HASH_COMMIT
- ☐ git reflog y luego git checkout HASH_COMMIT
- ☐ git restore <archivo>
- ☐ git reparo

Pregunta nº 12

1 de 1 puntos

¿Qué comando o comandos usaste en el paso 32? *

1/1

- ☒ git reflog y luego git reset --hard HASH_COMMIT
- ☐ git reflog y luego git checkout HASH_COMMIT
- ☐ git restore <archivo>
- ☐ git reducio

Pregunta nº 13

1 de 1 puntos

¿Qué comando o comandos usaste en el punto 33? *

1/1

- ☒ git reflog y luego git reset --hard HASH_COMMIT
- ☐ git reflog y luego git checkout HASH_COMMIT
- ☐ git restore <archivo>
- ☐ git imperio