

1.

Git funciona para muchos tipos de archivo, ¿Qué tal funciona Git con archivos binarios?

Los puede agregar y versionar, pero es mejor usar Git con texto plano.

2.

¿La terminal ideal para usar Git en Windows es?

Git Bash o una *terminal* derivada de Linux

3.

¿Qué guarda Git?

Los cambios de los archivos de un proyecto.

4.

En un commit, los mensajes del commit son:

Importantes y obligatorios.

5.

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

Un área temporal en memoria antes de llegar al repositorio.

6.

Con checkout puedo:

Ver todos los archivos de mi proyecto como los dejé en otras ramas.

7.

Un repositorio remoto y un repositorio local:

Tienen ramas similares dependiendo de cuáles haya enviado entre uno y otro.

8.

¿El HEAD en una rama es?

El apuntador al estado actual del repositorio basado en la rama en la que estoy trabajando.

9.

¿Cuándo debería hacer un merge?

Cuando quiero fusionar los cambios de una rama con otra.

10.

Cuando hay un conflicto entre archivos lo mejor es:

Analizar los conflictos tal como los reportó Git y elegir los cambios finales, luego commit.

11.

¿GitHub es?

Un sistema online de manejo de repositorios de Git.

12.

¿Las llaves públicas son?

Llaves que puedes compartir

13.

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

Usar una llave SSH.

14.

¿Qué crean los tags en Git?

Versiones descargables.

15.

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

Una rama independiente por cada persona y sus cambios que luego con verificación se hace merge a master.

16.

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

Editarlo, aceptarlo, darle merge, comentarlo para pedir cambios.

17.

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

Copiar un repositorio público a mis repositorios en GitHub, con todas sus ramas e historia anterior.

18.

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

```
<div id="post">
```

19.

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

Memoria temporal, volviendo al estado del último commit.

20.

¿Con amend puedo?

Corregir los mensajes de un commit que hice mal sin que quede en la historia del repositorio.

21.

¿Qué puedes hacer con gitk?

Ver en una interfaz visual las ramas y el log de tu repositorio.

22.

Si usas:

```
git config --global alias.platzi "shortlog"
```

¿Cómo invocar ese comando?

```
git platzi
```

23.

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

Debemos aprender Git con sus comandos de la terminal para resolver problemas o conflictos más avanzados. Gitk funciona bien, pero no nos permite realizar operaciones tan complejas.

24.

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

GitHub los llama Pull Request y GitLab Merge Request.

25.

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

Verdadero

26.

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

git rm saca los archivos del repositorio y (opcionalmente) del disco duro. git reset head saca los archivos de Staging, pero no del disco duro.

27.

¿Para qué sirve git grep?

Para encontrar las veces que hemos usado una palabra en los archivos del repositorio.

28.

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

Verdadero

29.

¿Para qué sirve GitHub Pages?

Es un servicio de GitHub que nos permite publicar nuestros repositorios en internet (por ejemplo, nombre.github.io o nombre.github.io/proyecto).

30.

El Jefe Freddy despidió a Anita y no piensa volver a contratarla. ¿Qué debe hacer Anita para seguir haciendo contribuciones a los proyectos públicos de Platzi?

Crear un Fork del proyecto y enviar Pull Request al proyecto original.