

SISTEMAS DE CONTROL DE VERSIONES



Alex Domínguez

1.- Un poco de historia: ¿Quién es el autor de Git? ¿Quién lo mantiene en la actualidad? ¿Qué tipo de licencia necesita? ¿En qué año se lanzó? ¿Cuál es la última versión estable?

El autor de Git fue Linus Benedict Torvalds, nacido en Finlandia en 1969 y lanzando su software de control de versiones el 7 de abril de 2005. Actualmente lo mantiene Junio Hamano apoyado por decenas de programadores y manteniéndolo como un software libre. Actualmente la versión estable más reciente es la 2.24.0, lanzada el día 4 de noviembre de 2019

2.- ¿Qué es un servidor de versionado remoto?

Es un servidor destinado a sostener un software de versionado al cual se pueda acceder desde otros equipos presentes en la misma red. Ya sea local o vía internet.

3.- Enumera servidores basados en Git.

GitHub, GitLab, Bitbucket, Gitea, Gogs, Phabricator, RhodeCode, Gitblit, Gerrit.

4.- Control de version con Git: Comparación de servidores remotos

Github frente a Gitlab y Bitbucket. Github es un repositorio remoto global, accesible desde internet. Por su parte, Gitlab y Bitbucket son repositorios remotos privados. Software que se instala en un servidor y del que solo se puede hacer uso si se está conectado al servidor anfitrión.

5.- Después de realizar la comparación. ¿Qué repositorio deberías escoger y por qué?

Depende de para que lo quiera usar. Si lo que quiero es trabajar un proyecto desde lugares diferentes o trabajar en equipo desde diferentes lugares. Lo más cómodo es Github. Si por el contrario, lo que busco es trabajar solo o en grupo en un único lugar estando siempre trabajando cara a un servidor de desarrollo. Lo más conveniente es usar un repositorio remoto propio del servidor sobre el que se trabaja.

6.- Resume y explica las operaciones con un servidor remoto.

Existen 4 acciones básicas a realizar cara a un servidor remoto (Push, Fetch, Pull y Clone). Push: Subir al repositorio todos los cambios realizados contenidos en el

último “commit” generado. Fetch: Descargar del repositorio la última versión de una rama. Pull: Descargar y mezclar con la rama activa la última versión de una rama del repositorio Clone: Descargar todo el contenido de un repositorio.

7.- ¿Qué tipo de repositorios nos permite crear Github? ¿Y Gitlab?

Mientras Github nos permite crear repositorios públicos (acceso para cualquier usuario en el mundo) y privados (Solo accesibles a quien les das ese privilegio). Gitlab nos permite esos anteriores además de los de tipo interno (Solo accesibles si estás registrado en la aplicación).

8.- Diferencia entre los comandos fetch y pull.

Si bien el comando fetch sirve para descargar la última versión de las ramas de un repositorio y almacenarlas en la parte remota del repositorio interno del equipo. Fetch hace lo mismo además de tratar de actualizar automáticamente la rama activa de desarrollo con esos datos. Se podría decir que pull es la suma de los comando fetch y merge.