QUÉ ES GIT Y CUALES SON LAS VENTAJAS

Git es un sistema de control de versiones usado en el desarrollo de software para guardar los cambios realizados en un proyecto, mantener un historial y navegar a través del mismo en cualquier momento.

Git es un sistema que se usa mediante la terminal de comandos.

**Windows**

Debido a que el Command Prompt (o consola de Windows) no es muy amigable y no soporta comandos de Unix, la web de [Git](https://git-scm.com/" \t "_blank) nos proporciona una herramienta muy buena llamada Git Bash y que nos permitirá escribir comandos como si estuviésemos en Linux o Mac OS X.

**OCHO RAZONES PARA USAR GIT**

Para los que aún no han oído hablar de [Git](https://git-scm.com/) (donde estuvieron?!), es un software de gestión de versiones diseñado por [Linus Torvalds](https://es.wikipedia.org/wiki/Linus_Torvalds), creador del kernel de Linux, para mantener de forma eficaz un proyecto de software de forma distribuida, ordenada y robusta.

Si todavía están usando FTP o SFTP para desplegar sus proyectos en un servidor, es hora de que reconsideren pasarse a Git, y abrazar todas las ventajas de la gestión de versiones

Veamos algunos de los problemas que surgen con este enfoque:

1- Trabajo colaborativo. Cómo hacemos para trabajar varios desarrolladores en un mismo archivo? Qué versión es la final? Cómo hacemos para que nuestros cambios no sobrescriban los de mis compañeros?

2- Volver a una versión anterior. Ctrl+Z es la única forma de volver atrás en un mismo archivo, eso teniendo en cuenta que el archivo o el editor tengan la historia del mismo, de lo contrario, estamos perdidos.  
Para evitar eso, tenemos varios backups del mismo archivo y sus distintas versiones editadas a lo largo del tiempo.

3- Subir múltiples archivos en un proyecto muy grande (1000+) lleva muchísimo tiempo, ya que generalmente reemplazamos **TODO** el proyecto, o si no tenemos que ir manualmente, directorio por directorio, subiendo sólo los archivos cambiados.  
Lento y propenso a errores, además del ancho de banda que consume reemplazar todo el proyecto cada vez que hacemos cambios.

4- Facilidades a la hora de compartir el código. Supongamos que queremos hacer una librería o un proyecto **Open Source**. Quienes tienen permiso a que archivos, cómo nos ponemos de acuerdo para editar código sin pisar el trabajo del otro, cómo evitamos que se borre o pierda parte del trabajo por algún error?

Si todavía están usando FTP o SFTP para desplegar sus proyectos en un servidor, es hora de que reconsideren pasarse a Git, y abrazar todas las ventajas de la gestión de versiones

4- **Permite generar flujos de trabajo** que facilitan el desarrollo y mantenimiento de proyectos de gran tamaño.

5- **El ecosistema Git es increíble**, y agrega un montón de herramientas a nuestra disposición para facilitarnos el trabajo, de forma robusta, rápida y profesional.

A través de los **"hooks"** de Git, los distintos servicios pueden detectar cambios en el historial de versiones y realizar acciones automáticas (como actualizar los archivos en el servidor o ejecutar una suite de tests y enviarnos su resultado), dejándonos tiempo libre para cosas más productivas!

6- Las **"branches"** o ramas, permiten trabajar con una base de código paralela al proyecto en sí, donde podemos corregir bugs o desarrollar nuevas características para el producto sin afectar el **"master"**, pero manteniendo todas las ventajas de usar un sistema de control de versiones. Una vez que estamos contentos con nuestro **"branch"**, podemos combinarlo con el **"master"** o, en lenguaje Git, hacer un **"merge"**.

7- Empezar a trabajar desde otro entorno es tan fácil como **"clonar"** el proyecto a tu nuevo entorno, trabajar sobre los archivos que se quieran, y subir los cambios al **"master"** o a una **"branch"**.

8- Sistema de etiquetas, para etiquetar las distintas versiones del proyecto. Esto es un marcador a una versión específica del proyecto, sólo que en lugar de tener distintos backups de versiones anteriores, apuntamos a distintas versiones dentro de la misma base de código.

Y si todo eso no los entusiasma para cambiarse a Git, les digo que por experiencia personal, una vez que empiecen a usarlo, nunca más van a poder trabajar en un proyecto en el que no lo tengan!

Café de por medio / un blog sobre desarrollo de Software. (16 de julio de 2016) 8 razones para usar Git. Recuperado de: <https://blog.coffeedevs.com/8-razones-para-usar-git/>