¿Qué es GIT?

Antes de comenzar, me gustaría aclarar una diferencia de conceptos.

GIT y GItHub son totalmente diferentes. GIT es el software y GitHub es la comunidad que desarrolla con GIT. Son complementos, pero personajes independientes.

¿Qué es GIT?

Es un software rastreador. **Le da seguimiento a todos los cambios que se ejecutan sobre un archivo o carpeta.** Cada cambio que hagas en un directorio, GIT se da cuenta y lo registra. Así de simple.

Imaginemos un archivo que se modifica constantemente:

* Index.html
1° cambio: Agregamos doctype
2° cambio: Agregamos head
3° cambio: Agregamos body
...

Cada vez que haces un cambio en tu código, **GIT registra los cambios y los guarda.** Se te ofrecerá acceso al autor, fecha, qué se modificó exactamente y comparación de cambios.

Si hago 10 cambios, GIT guarda 10 veces todos mis archivos? No estaría generando miles de archivos?

Este concepto y pregunta es muy normal. **GIT guarda los cambios que haces, no las versiones.**

GIT no clona 10 veces tu proyecto, sino que registra cuáles fueron las líneas que modificaste, las encapsula en un registro que se llama

“commit” (lo veremos más adelante) y con esto, te permite disfrutar de un historial de avances de tu proyecto sin preocuparte por el peso, revisando todos los "commits" hechos.

Características de GIT

a) Es un sistema de control de versiones distribuido. Con esto, nos referimos a que GIT clona los proyectos para que cada persona ó miembro de un equipo tenga una copia exacta y completa de todo el código, historial y las personas que estuvieron involucradas, también **conocido como repositorio.**

Si se llegase a perder el repositorio original, no habría mucho drama porque es probable que existan personas que tienen un clon y se puede partir desde ahí sin problema.

Básicamente, **cada persona (o grupo de personas) mantienen y trabajan sus propios repositorios,** derivados del principal, el cual, con toda la flexibilidad, se pueden fusionar y compartir avances.

Visualicemos un repositorio y sus clones:

```
Repo Original: a, b, c, d
```

Cada letra (a,b,c...) se refiere a un cambio del proyecto (un estilo, una línea HTML, alguna función de JS, etc.). Vemos como hay 4 cambios: a,b,c,d.

Se generará 1 clon del repositorio, perteneciente a otra persona que quiere colaborar:

```
Repo Original: a - b - c - d
| - Repo Clon: a - b - c - d
```

Este nuevo clon contiene el mismo registro de cambios, archivos e historial del repositorio original. En este momento, cada persona puede seguir avanzando su propio proyecto con sus respectivos cambios:

```
Repo Original: a - b - c - d - e - f
| - Repo Clon: a - b - c - d - x - y - z
```

Como podemos observar se harán diferentes desarrollos en cada repositorio. **Ninguno es malo ni bueno, simplemente se trabaja independiente.** La idea es que no son dependientes pero tendrán la oportunidad de sincronizarse en el momento que gusten.

Si llegan a aparecer más repositorios clones, también pueden colaborar entre ellos sin depender del repositorio principal.

Claro está, que **es importante siempre estar actualizado con el repositorio original**, ya que el autor siempre estará gestionando que el proyecto se encuentre estable y mejorándose.

A esto nos referimos con un sistema de control de versiones "distribuido".

b) Es Open Source. GIT no cuesta, puedes instalarlo en cualquier ordenador o servidor.

c) Colaborativo. Si tienes un proyecto y compartes el código, las personas interesadas o que forman parte de tu equipo pueden agregar nuevas características, arreglar bugs o comentar.

Descargar GIT

Para poder instalar GIT en tu computadora, lo primero que tienes que hacer es entrar a esta página:

Website Oficial GIT

Nos vamos a la sección "Downloads" y escogemos nuestro sistema operativo.

Algo muy importante es que se está instalando **GIT para trabajar en consola, NO estamos instalando los clientes** (el cual incluyen interfaces de usuario amigables para gestionar los proyectos).

Para trabajar con los clientes de GIT (incluidos los de Github), hay una sección llamada "GUI Clients", ahí ustedes pueden descargar los clientes para una mejor gestión de su código sin utilizar supuestamente la terminal.

"Odio la terminal..."

Recomendamos mucho aprender a utilizar GIT en la terminal debido a que los clientes (GUI's) pueden en algún momento tener un conflicto entre archivos y se le pide al usuario que lo resuelva a través de la terminal. La terminal es tu amiga.

Es como aprender a manejar un auto en automático y estándar. Si te vas directo al automático, pierdes la oportunidad de manejar otro tipo de vehículos, uno nunca sabe cuando necesitarás la habilidad de utilizar la palanca de velocidades.

¿Está GIT instalado?

Para confirmar que GIT esté totalmente listo para trabajar, abrimos nuestra terminal ó consola y después ejecutamos el siguiente comando:

```
$git --version
 git 1.8.0.3
```

Ya estás listo para iniciar con GIT.

Configuremos y personalicemos GIT

Una vez instalado, debemos de configurar GIT. ¿Por qué configurarl? Sencillo, vamos a decirle con qué nombre y correo vas a utilizar GIT, además de establecer una conexión de tu computadora a GitHub, así podrás colaborar y descargar proyectos más fácilmente.

```
$git config --global user.name "Miguel Nieva"
$git config --global user.name "m@mejorando.la"
```

El comando "config" se utilizará para la configuración de GIT. Usamos "global" para que tengamos todos los permisos.

"user.name" que significa el nombre usuario y "user.email" que significa el correo electrónico (es importante que coloques el correo electrónico que usas con Github para que se vincule con la cuenta).

¿Listo? Perfecto, vamos a verificar que todo este correcto. Introduzcamos:

```
$git config --list
user.name=Miguel Nieva
user.email=m@mejorando.la
```

Genial, ya tienes Git instalado y enlazado con GitHub. Te quedaste con dudas? Al sistema de discusiones. En el siguiente material aprenderemos a usarlo.