#### **Git**

Uso de Github

Owner - Colaboradores

Repositorios

Proyectos

Issue

Cards

Control de Versiones

Volver a una version anterior

Ejemplos

Repositorios de

Android

PHP

MVC

#### 

#### 

#### 

#### **-----------------------------------**

#### **¿Pero qué es** [**Git**](http://git-scm.com/)**?**

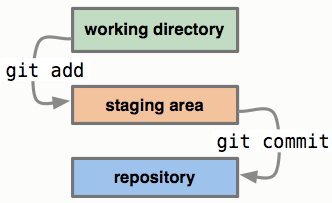
[**Git**](http://git-scm.com/)es un Sistema de Control de Versiones (*VCS)* distribuido [**libre y de Código Abierto**](http://www.opensource.org/docs/osd)escrito en lenguaje C, *¿Qué quiere decir esto?,* Que si tu y un grupo de amigos trabajan en un proyecto de desarrollo en cualquier lenguaje(s) del lado del cliente (*Frontend*) o del servidor (*Backend*), incluso de ambos tipos, pueden interactuar con el código del proyecto, ¡todos!, sin la necesidad de tener explícita y obligatoriamente un servidor.Pero! Si es recomendable así sea un sistema distribuido; tener un repositorio central (Para esto usaremos [**GitHub**](https://github.com/)).

#### **Vocabulario en** [**Git**](http://git-scm.com/)

Es necesario tener un vocabulario o una jerga básica referente a Git, una lista con algunos conceptos.

|  |  |
| --- | --- |
| **Termino** | **Definición** |
| Repositorio | Un repositorio contiene la historia, las diferentes versiones en el tiempo y todas las diferentes ramas *(branches)* y etiquetas *(Tags)*. Cada copia del repositorio en Git es un repositorio completo. |
| Ramas ó Branches | Una rama o branch es una línea de código por separado con su propia historia. Puede crear una nueva rama de una existente y cambiar el código de forma independiente de otras ramas. Una de las ramas es el valor por defecto (normalmente llamado *master* ó maestro). |
| Etiquetas ó Tags | Una etiqueta ó Tag apunta específicamente a cierto espacio en el tiempo en una rama ó branch específico. Con un tag, puede tener un punto de referencia al que siempre puede revertir, por ejemplo, el código del 15.05.2012, en el branch *“test”* o el código de la versión 1.5 en el brach*“master”*. |
| Commit | Difícilmente en español se pueda traducir esta palabra, pero es como confirmar cambios. Esto crea una nueva revisión, que puede ser recuperada más tarde, por ejemplo, si usted quiere ver o recuperar el código fuente de una versión anterior. Cada commit contiene el autor y el committer o quien realiza los cambios, por lo que es posible identificar la fuente del cambio. El autor y el committer podrían ser personas diferentes. Esto lo veremos más adelante. |
| Revisión | Representa una versión del código fuente. Git identifica las revisiones con los identificadores de [SHA1](http://www.sha1.cz/). Los [SHA1](http://www.sha1.cz/) son los identificadores de 160 bits de longitud y están representados en hexadecimal. La última versión se puede abordar a través de “HEAD”, la versión antes de que a través de “HEAD ~ 1” y así sucesivamente. |

* ***El directorio de Git*** es donde se almacena los metadatos y la base de datos de objetos para tu proyecto. Es la parte más importante de [**Git**](http://git-scm.com/), y es lo que se copia cuando clonas un repositorio desde otro ordenador, puede ser desde el servidor, de esto hablaremos en el siguiente o posterior Post.
* ***El directorio de trabajo*** es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de [**Git**](http://git-scm.com/), y se colocan en tu disco local para que los puedas usar o modificar, esto ocurre cuando clonamos o creamos un proyecto.
* ***El área de preparación*** es un sencillo archivo, que se encuentra en la mitad de la grafica y generalmente contenido en tu directorio de [**Git**](http://git-scm.com/), que almacena información acerca de lo que va a ir en tu próxima confirmación. Llamado en algunos casos como el índice, y se está convirtiendo en estándar el referirse a ello como el área de preparación.



Pero no te preocupes que en el siguiente Post con ejemplos y ejercicios aplicaremos estos conceptos, por el momento construya un esquema mental con esta información. Ahora teniendo en cuenta estos tres estados el flujo de trabajo básico en Git es algo así:

1. Modificas una serie de archivos en tu directorio de trabajo.
2. Preparas los archivos, añadiendo instantáneas de ellos a tu área de preparación, ósea tu carpeta local de trabajo.
3. Confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación, y almacena esa instantánea de manera permanente en tu directorio de Git.