

Control de versiones y Git

Ernesto Rodriguez

Universidad del Itsmo

erodriguez@unis.edu.gt

- Es difícil juntar el código de diferentes personas.
- Es importante tener un registro de todos los cambios que ha tenido un programa.
- Es difícil determinar cuando y quien introdujo un error.
- Es conveniente poder comparar diferentes versiones de un programa.
- Es importante proteger el código fuente ante la introducción de errores debido a fallos en sistemas.

En resumen, el control de versiones introduce orden y estructura en el desarrollo de un programa, lo cual permite que miles de personas en todas partes del mundo con diferentes habilidades trabajen en un proyecto. Como por ejemplo: Linux.

- El proceso de llevar un registro de todos los cambios que han habido en un repositorio.
- Un *repositorio* es un conjunto de archivos y recursos que confirman algun proyecto. Generalmente, un proyecto de software.
- El contenido de un repositorio se representa mediante revisiones.
- Un *controlador de versiones* es una herramienta que permite llevar el control de versiones.
- Existen varios controladores de versiones: Git, Subversion, Mercurial
- Github es un servicio para alojar repositorios.

- Una revision es una descripción de modificaciones que han ocurrido en un repositorio.
- Las revisiones estan ordenadas secuencialmente.
- El contenido de un repositorio se construye aplicando revisiones en sequencia.

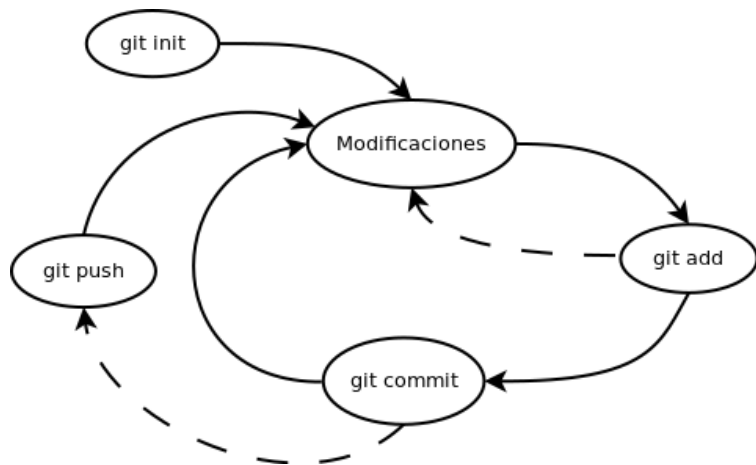
$$\text{contenido}_N = \text{revision}_1 \oplus \text{revision}_2 \oplus \dots \oplus \text{revision}_N$$

Comandos basicos de Git

<code>git init</code>	Crea un repositorio en la carpeta actual.
<code>git status</code>	Listar las modificaciones actuales.
<code>git add (lugar1) [lugar2 ...]</code>	Agregar archivos a la siguiente revision.
<code>git commit [-m "descripcion"]</code>	Crear una nueva revision.
<code>git help [commando]</code>	Obtener ayuda sobre un comando.
<code>git push</code>	Publicar cambios en otro repositorio.

Para mayor información, ver el tutorial de Git[1].

Ciclo de vida de Git



- Archivo especial en formato *glob*[2].
- Permite mantener el repositorio minimalista al ignorar archivos generados automaticamete.
- Ejemplos de archivos ignorados:Codigo de maquina (*.dll).
- Este repositorio contiene un archivo *.gitignore* en su carpeta base.

 Software Freedom Conservancy.
Git.
<https://git-scm.com/docs/gittutorial>.

 Software Freedom Conservancy.
Ignoring files.
<https://git-scm.com/docs/gitignore>.