

UNIVERSIDAD DON BOSCO



DESARROLLO DE APLICACIONES WEB CON SOFTWARE INTERPRETADOS
EN EL CLIENTE

DOCENTE

JONATHAN ESCALANTE

GRUPO 02T

CICLO 02/2020

“INVESTIGACIÓN APLICADA 1”

ESTUDIANTE: RAMOS JIMÉNEZ, JAIRO DENNIS

RJ172021

SOYAPANGO, 01 DE SEPTIEMBRE DE 2020

SISTEMA DE CONTROL DE VERSIONES

El control de versiones son una categoría de herramientas de software que nos permite gestionar los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo de manera que sea posible recuperar versiones específicas más adelante. Si se comete un error, los desarrolladores pueden ir atrás en el tiempo y comparar las versiones anteriores del código para ayudar a resolver el error al tiempo.

Estos se utilizan en entornos de desarrollo de software, pero puede resultar de gran ayuda para cualquier persona que necesite un control sobre la tarea que está realizando. Un ejemplo más claro de estos es al editar cualquier tipo de archivo en servicios de almacenamiento en la nube como Google Drive o Dropbox.

Para la mayoría de equipos de software, el código fuente es “un repositorio del conocimiento de valor incalculable y de la comprensión sobre el dominio del problema que los desarrolladores han recopilado y perfeccionado con un esfuerzo cuidadoso”. El control de versiones protege el código fuente tanto de las catástrofes como del deterioro casual de los errores humanos y las consecuencias accidentales.

Los desarrolladores de software que trabajan en equipos están escribiendo continuamente nuevo código y modificando el ya existente. El código de un proyecto, una página web o una aplicación se organiza en una estructura de carpetas o "árbol de archivos". Un desarrollador del equipo podría estar trabajando en una nueva función mientras otro desarrollador soluciona un error no relacionado cambiando código. Cada desarrollador podría hacer sus cambios en varias partes del árbol de archivos.

El control de versiones ayuda a los equipos a resolver estos problemas al realizar un seguimiento de todos los cambios individuales de cada desarrollador y evitar que el trabajo concurrente entre en conflicto. Los cambios realizados en una parte del software pueden ser incompatibles con los que ha hecho otro desarrollador que está trabajando al mismo tiempo. Este problema debería detectarse y solucionarse de manera ordenada sin bloquear el trabajo del resto del equipo. Además, en todo el desarrollo de software, cualquier cambio puede introducir nuevos errores por sí mismo y el nuevo software no es fiable hasta que se prueba. De este modo, las pruebas y el desarrollo van de la mano hasta que está lista una nueva versión.

VENTAJAS DE LOS SISTEMA DE CONTROL DE VERSIONES

Desarrollar software sin utilizar el control de versiones no es recomendable ya que no se tendrían copias de seguridad. El control de versiones también puede permitir que los desarrolladores se muevan más rápido y posibilita que los equipos de software mantengan la eficacia y la agilidad a medida que el equipo se escala para incluir más desarrolladores.

Los sistemas de control de versiones han experimentado grandes mejoras en los últimos años. Una de las herramientas más populares hoy en día se llama Git. Git es un sistema de control de versiones, una herramienta para administrar el historial del código fuente. Al igual que muchos de los sistemas más populares disponibles hoy en día, Git es gratuito y de código abierto. Independientemente del nombre que tengan, o del sistema que se utilice, las principales ventajas que deberías esperar del control de versiones son las siguientes:

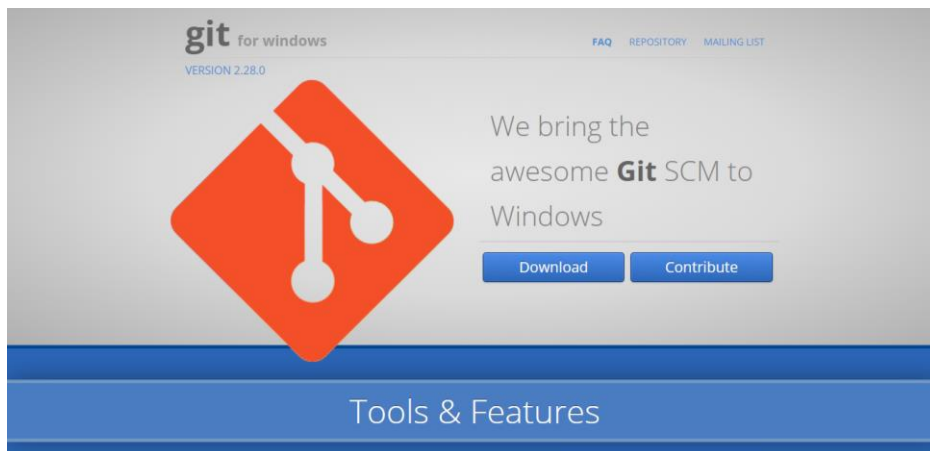
- Un completo historial de cambios a largo plazo de todos los archivos. Tener el historial completo permite volver a las versiones anteriores para ayudar a analizar la causa raíz de los errores y es crucial cuando se tiene que solucionar problemas en las versiones anteriores del software.
- Muestra mucha información estadística de cómo se está desarrollando el proyecto (principales autores, número de versiones, cambios, etc.).
- Creación de ramas y fusiones. La creación de una rama mantiene múltiples flujos de trabajo independientes los unos de los otros al tiempo que ofrece la facilidad de volver a fusionar ese trabajo, lo que permite que los desarrolladores verifiquen que los cambios de cada rama no entran en conflicto.
- Permite mantener un histórico de todo el desarrollo del proyecto.
- Se tiene la posibilidad de ver qué cambios se han hecho en el código en cada versión. Tener el historial anotado del código a tu alcance cuando estás leyendo el código, intentando entender lo que hace y por qué se ha diseñado así, puede permitir a los desarrolladores hacer cambios correctos y armoniosos que estén en línea con el diseño previsto a largo plazo del sistema.

Aunque se puede desarrollar software sin utilizar ningún control de versiones, hacerlo somete al proyecto a un gran riesgo que ningún equipo profesional debería aceptar.

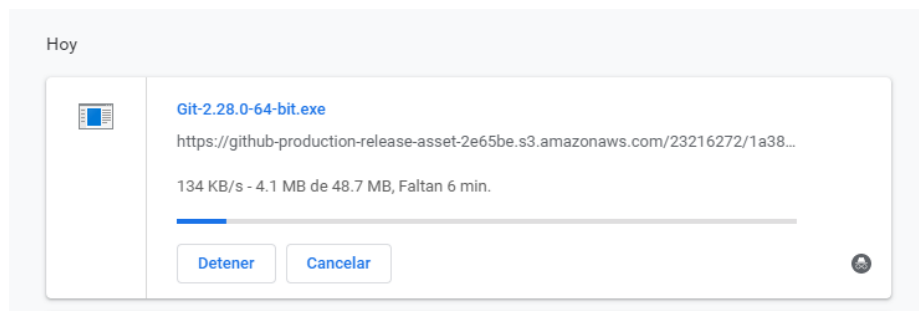
GUÍA DE INSTALACIÓN PARA GIT (WINDOWS)

Para empezar a utilizar Git tienes que instalarlo en tu computadora primero. Incluso si ya está instalado, es recomendable actualizarlo a su última versión. Puedes instalarlo como un paquete, a partir de un archivo instalador o bajando el código fuente y compilándolo tú mismo.

1. Abre cualquier navegador y accede a la siguiente dirección: <http://msysgit.github.com>



2. Entra en la sección *Downloads* y descarga la versión más reciente



3. Luego de descargarlo instala el archivo **.exe** como cualquier otra aplicación de Windows.



Configuración

Comprueba que la instalación ha sido correcta abriendo una nueva consola de comandos y ejecutando el siguiente comando sin argumentos:

```
> git
```

Como resultado deberías ver el listado de comandos Git disponibles:

```
usage: git [--version] [--exec-path[=<path>]] [--html-path] [--man-path]
        [--info-path] [-p|--paginate|--no-pager] [--no-replace-objects]
        [--bare] [--git-dir=<path>] [--work-tree=<path>]
        [--namespace=<name>] [-c name=value] [--help]
        <command> [<args>]
```

BIBLIOGRAFÍA

Bitbucket. (2020). *Bitbucket*. Obtenido de <https://bitbucket.org/product/es/version-control-software>

Swift by Coding. (2019). *Swift by Coding*. Obtenido de <https://swiftbycoding.dev/git/sistemas-de-control-de-versiones/>