Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Maestría en Ciencias de la Computación
Fundamentos de Programación y Scripting “A”

Nombre: Francisco Javier Hernández Azurdia
DPI: 3015511460101
Carné: 999012591

GIT

¿Qué es GIT?

GIT es un software denominado como Sistema de Control de Versiones (VCS), el cual registra todas las versiones de un proyecto, con la finalidad de tener la capacidad de consultar los cambios o modificaciones realizados al proyecto de manera ordenada y poder regresar a versiones anteriores en el caso se produce un error a la hora de programar. Los proyectos se manejan en un directorio principal conocido como repositorios, en los cuales se genera una línea de tiempo de información con registros o *snapshots* de las modificaciones realizadas en el proyecto, sin necesidad de la generación de copias del mismo archivo.

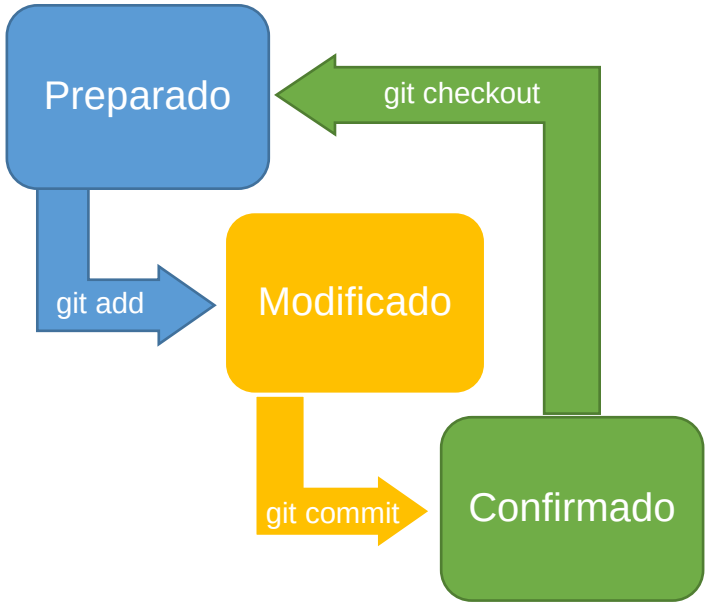
Control de versiones con GIT

El control de versiones en GIT se diferencia a un control de versiones manual en el escenario en el cual, si es necesario consultar o regresar a una versión anterior, se consulta directamente en el historial de registros realizado en lugar de consultar una copia anterior guardada por el usuario. GIT permite realizar un control de versiones distribuido, el cual consiste en tener la posibilidad de trabajar un proyecto con diversos colaboradores de forma paralela mediante un clon del repositorio, creando un respaldo de la información y un historial de cambios realizados. Mediante el uso de ramas o *branchs*, es posible que todo colaborador pueda trabajar de manera independiente en un rol predeterminado de manera ordenada.

Estados de un archivo en GIT

GIT administra todas las versiones de un proyecto en tres estados: preparado, modificado y confirmado. En el estado preparado se encuentran todos los cambios realizados en un proyecto, ya sea la adición de un archivo o el cambio en la programación realizada. En el estado modificado se encuentran los archivos que se encuentran preparados para ser incorporados en una versión del proyecto. En el área de confirmado se crea una versión del proyecto con todas las modificaciones

del proyecto, la cual puede ser consultada por el usuario o por terceros que tengan acceso al repositorio.



¿Cómo se configura un repositorio?

Existen dos maneras de obtener un repositorio de GIT, crear un repositorio desde cero o clonar un repositorio existente. A partir de la creación de un nuevo proyecto, mediante el comando **git init,** es posible incorporar GIT, lo cual genera un subdirectorio .git en el directorio actual, además de generar una rama principal. Si se cuenta con un proyecto previamente creado, se ejecuta el comando **git init** en el directorio de la carpeta raíz del proyecto.

En el caso se desee obtener un repositorio de un tercero o bien, un repositorio propio ubicado en la web es posible clonar el repositorio en un ordenador, con el comando **git clone**, seguido de una dirección URL, en la cual se encuentre el repositorio. Esto permite acceder a una copia de un proyecto y a todas sus versiones de forma local, además de agregar una rama principal nueva.

A la hora de realizar un cambio en el repositorio y desear generar una nueva versión, se emplean los comandos **git add** para preparar una versión y el comando **git commit** para que esta sea publicada.

Comandos en GIT

- **git add:** Prepara los cambios realizados para la creación de una versión de un proyecto.
- **git branch:** Crea una rama o *branch*, la cual permite organizar un proyecto en divisiones utilizadas por diferentes colaboradores.
- **git checkout:** Permite navegar en ramas existentes y extraer versiones antiguas de un proyecto.
- **git clone:** Realiza una copia de un repositorio externo en un dispositivo local.
- **git commit:** Crea una versión de un proyecto con los cambios realizados.
- **git fetch:** Descarga una rama de otro repositorio sin combinar los archivos al repositorio local.
- **git init:** Permite crear un nuevo repositorio de GIT.
- **git log:** Permite observar todas las versiones de un proyecto.
- **git merge:** Combina los cambios realizados en un proyecto en todas las ramas del repositorio.
- **git pull:** Descarga y combina una rama o versión al proyecto local.
- **git push:** Publica la última versión generada para que pueda ser vista por los integrantes del repositorio.
- **git rebase:** Ayuda a administrar el orden de las ramas de un repositorio.
- **git remote:** Emplea un método más eficaz para el uso del repositorio de forma remota.
- **git reset:** Borra los cambios realizados en el proyecto que no han sido publicados en una versión.
- **git revert:** Borra una versión del proyecto ya publicada.
- **git status:** Permite visualizar los cambios preparados por **git add** y **git commit** antes de que estos sean publicados.

Bibliografía

- ATLASSIAN. (s.f.). Glosario git. Obtenido de ATLASSIAN Bitbucket: https://www.atlassian.com/es/git/glossary
- EDteam. (25 de Agosto de 2019). *¿Qué es Git y cómo funciona? - La mejor explicación en español*. Obtenido de EDteam YouTube: https://www.youtube.com/watch?v=jGehuhFhtnE&ab_channel=EDteam
- Fiqus LTDA. (Abril de 2016). *Git - Manual de Usuario Version 1.* Obtenido de Fiqus: https://www.valor.es/wp-content/uploads/2016/04/git_manual.pdf
- Gómez Bachiller, S. (Noviembre de 2015). *Introducción a GIT Y GITHUB - DÍA 1.* Obtenido de Aula Software Libre:

https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf

- SProgramación. (31 de Diciembre de 2019). *Git y Github #3 Estados de Git*. Obtenido de SProgramación YouTube: https://www.youtube.com/watch?v=i285V5sZSmQ&t=72s&ab_channel=SProgramacion