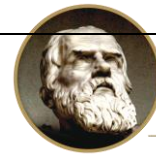


Git & GitHub

César Rodríguez, 22001035

Guatemala 03 de febrero del año 2023



Contenido

¿Qué es GI?	3
Conceptos esenciales:	3
Ramas:.....	3
Principales Ventajas de utilizar la solución de Git.....	5
¿Qué es GitHub?	6
Markdown.....	6
Comandos básicos de Git en consola	6
Referencias.....	7

¿Qué es GI?

Se le conoce por ser un sistema de control de versiones distribuido, siendo un repositorio de control de versiones completo. De tal forma que los desarrolladores confirman su trabajo localmente y, posteriormente sincronizan su copia del repositorio con la copia en el servidor.

Conceptos esenciales:

Cada vez que se guarda un trabajo en particular, Git crea una confirmación, siendo esta una instantánea de todos los archivos en un momento dado. Mientras que una archivo no haya cambiado de una confirmación a la siguiente, siempre se tomará el archivo almacenado anteriormente.

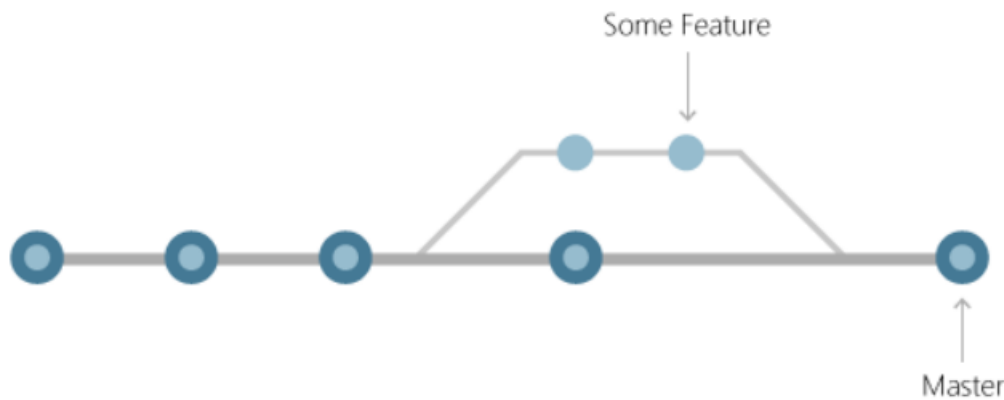
Las confirmaciones crean vínculos a otras confirmaciones, formando un gráfico del historial de desarrollo.



Obtenida de: Microsoft Learn

Ramas:

Git proporciona herramientas para aislar los cambios y volver a combinarlos posteriormente, permitiendo a los desarrolladores guardar los cambios en su propio repositorio de código local. Las ramas son punteros para trabajar en curso, las cuales administran una separación. Una vez finalizado el trabajo creado en una rama, se puede combinar de nuevo en la rama principal del equipo.

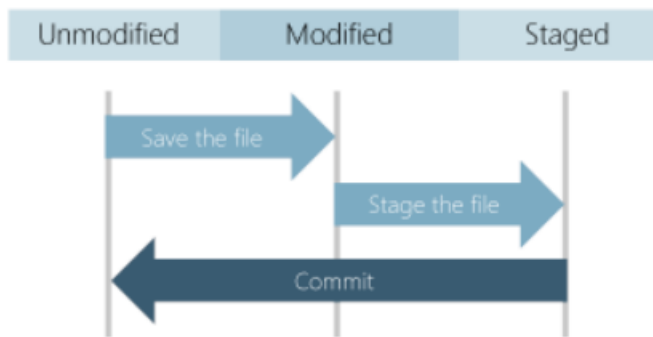


Obtenida de: Microsoft Learn

Archivos y Confirmaciones: Los archivos poseen 3 estados:

1. Modificados
2. Almacenados provisionalmente
3. Confirmados

Cuando se modifica un archivo por primera vez, los cambios solo existen en el directorio de trabajo. Todavía no forman parte de una confirmación o del historial de desarrollo. El desarrollador debe almacenar provisionalmente los archivos modificados que se incluirán en la confirmación. El área de almacenamiento provisional contiene todos los cambios que se van a incluir en la siguiente confirmación. Una vez que el desarrollador esté satisfecho con los archivos almacenados provisionalmente, los archivos se empaquetan como confirmación con un mensaje que describe lo que ha cambiado. Esta confirmación forma parte del historial de desarrollo.



Obtenida de: Microsoft Learn

Principales Ventajas de utilizar la solución de Git

1. Desarrollo simultáneo: Los usuarios poseen su propia copia local de código y pueden trabajar simultáneamente en sus propias ramas. Git funciona sin conexión, dado que casi todas las operaciones son locales.
2. Versiones rápidas: Las ramas permiten al desarrollador un modelo flexible y simultáneo.
3. Integración Integrada: Integra diversidad de productos y herramientas
4. Soporte técnico de la comunidad
5. Funciona en cualquier equipo
6. Solicitud de incorporación de cambios
7. Directivas de ramas: A través de Teams puede configurar GitHub y Azure DevOps para aplicar flujos de trabajo y procesos coherentes en todo el equipo

¿Qué es GitHub?

Solución en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Su potencial es inigualable dado que permite a desarrolladores colaborar y realizar cambios en proyectos compartidos mientras poseen un seguimiento detallado de su progreso.

Una de sus fortalezas es el control de versiones dado que su sistema permite rastrear y gestionar los cambios realizados en un archivo o conjunto de archivos. En su mayoría es utilizado por ingenieros de software para brindar un seguimiento oportuno de las modificaciones realizadas en el código fuente, el sistema de control de versiones les permite analizar todos los cambios y revertirlos sin repercusiones si se comete un error.

Markdown

RMarkdown y GitHub se caracterizan por utilizar la sintaxis de Markdown para reproducir fácilmente formatos de escrituras, un ejemplo es a través de HTML. Permite escribir un documento estructurado mediante elementos básicos para crear un texto: títulos, subtítulos, listas, letras en cursivas o en negritas, citas, entre otros.

(Said, 2019)

Comandos básicos de Git en consola

1. **git init** crea un nuevo repositorio Git.
2. **Git Status** inspecciona el contenido del directorio de trabajo y del área de ensayo.
3. **Git Add** agrega archivos del directorio de trabajo al área de ensayo.
4. **git diff** muestra la diferencia entre el directorio de trabajo y el área de ensayo.
5. **Git Commit** almacena permanentemente los cambios de archivo del área de ensayo del repositorio.
6. **Git log** muestra una lista de todas las confirmaciones anteriores.

Referencias

- B., G. (2020). *Hostinger Tutoriales*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-github>
- Cobs, M. (2022). *Microsoft Learn*. Obtenido de <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>
- Said, P. (2019). Obtenido de https://patosaid.github.io/apuntes_git_rmarkdown/#::~:~:text=Las%20sintaxis%20presentadas%20son%20algunas%20que%20se%20pueden,consola%20git%20init%20creates%20a%20new%20Git%20repository.