

# ¿Qué es Git y Github?

Didier Posse

*Departamento de Ingeniería Electrónica*

*Universidad Santo Tomás*

Bogotá, Colombia

didierposse@gmail.com

## I. GIT

### A. ¿Qué es un VSC?

En la página oficial de Git describen que este es un Sistema de Control de Versiones (VSC), la pregunta que uno se hace es ¿Qué es un VSC? Un VSC es un sistema que registra los cambios hechos en un archivo o un conjunto de archivos, esto quiere decir que uno puede guardar cada versión de los archivos a medida que uno los va modificando, esto es útil ya que si uno quiere recuperar el archivo en un estado anterior por si tiene algún problema que no quieres en él, puedes revertirlo fácilmente, también puede significar que si tiene alguna falla o se perdió el archivo, se puede recuperar fácilmente y sin necesidad de gastar tantos recursos. Otra ventaja es que puedes revertir el archivo si alguna otra persona tiene acceso al archivo, lo modificó y no te gusta cómo está.

### B. ¿Cómo funciona?

En Git, este sistema se divide en tres formas de guardar las versiones de los archivos, estos se crearon a medida que se daban cuenta de que los métodos sencillos de guardado eran propensos a errores que se podían solucionar, estos son las tres formas de guardado según la ocasión:

1) *Sistema de Control de Versiones Local*: Un método sencillo de control de versiones es copiar los archivos a otro directorio, el problema de esto es que al ser varios directorios uno fácilmente se puede confundir de directorio y escribir en un directorio equivocado. El sistema RCS opera colocando las diferencias que tienen las versiones del archivo en el disco, pero con un formato específico, y al agregar esas diferencias, se puede recrear cómo se veía cada versión del archivo.

2) *Sistema de Control de Versiones Centralizados*: Este funciona con un servidor central donde se almacenan los archivos con sus versiones y simplemente extraen los archivos desde ese servidor, esto hace que sea más fácil saber cómo se organizan las personas para modificar cada archivo en conjunto, es mejor que cada quien tenerlo localmente. Lo malo es que uno depende del estado del servidor, ya que si se cae el servidor o se llega a dañar parte de él, puede que no se pueda guardar por un tiempo el archivo o se pierda absolutamente todo, pero es poco probable por el mantenimiento que les tienen.

3) *Sistemas de control de versiones distribuidos*: El sistema distribuido tiene una ventaja muy grande y es la de almacenar los archivos de una mejor manera, este funciona con que se

puede guardar con varias copias de seguridad pero de una forma como si fuera por Torrent, el archivo se almacena en un servidor y también en los computadores que tengan el archivo, o sea, se copian unas clonaciones del archivo y si se daña o se borra, están los computadores de los colaboradores para poder recuperarlo.

### C. GitHub

GitHub es una plataforma que es utilizada para poder colaborar con otras personas y hacer trabajos entre sí, se pueden almacenar y compartir estos archivos, los códigos se almacenan en un repositorio y los repositorios pueden ser públicos o privados, entonces también es una plataforma de búsqueda que puede ayudar para aprender o tener parte de códigos de otras personas.

Esta plataforma también tiene función como una red social, uno se puede comunicar con cualquier otro programador, ver su perfil, ver sus estadísticas y los proyectos que ha desarrollado. Tiene colaboración con Git, lo cual tiene un Sistema de Control de Versiones para los archivos que uno esté desarrollando con los colaboradores.

## II. README

### A. ¿Qué es?

Es un repositorio de GitHub el cual te permite editar tu perfil de GitHub por medio de código, los códigos que soporta son el markdown y el HTML 5, esto permite una gran variedad de configuración del perfil, para que el perfil de cada quien sea original y único.

### B. Paso a paso

1) *Primero*: Uno crea una cuenta en GitHub, pone foto de perfil y si uno quiere configurar la cuenta como desee, simplemente se va al icono del perfil y le aparecen las opciones.

2) *Segundo*: Ya con la cuenta, uno crea un nuevo repositorio, rellena los campos que le piden rellenar para crear un repositorio, entre esas opciones uno coloca una visibilidad pública y activa la opción de "Add README".

3) *Tercero*: Teniendo ya el repositorio, uno tiene la opción de previsualización, de código y de Blame, la del Blame sirve para ver en qué día se hizo tal parte del código. Los códigos que se pueden compilar para el README son markdown y HTML 5, yo usé HTML 5 porque tiene más funciones para poder diseñar el perfil, usar los dos a la vez no me compilaba bien, así que por eso solo usé HTML 5.

4) *Cuarto:* Con estos conocimientos, uno ya puede codificar en la sección de código. Al principio puse como un título saludando a los que entren a mi perfil, y agregué un gif igual como se pone una imagen en HTML, para que se vea bonito, tipo wallpaper. Después quise poner los lenguajes de programación que me sé, pero de una forma original, así que los puse en una tabla horizontalmente, busqué iconos y los puse en cada sección de la tabla y abajo el nombre del lenguaje de programación. Como es HTML, solo necesitaba el link de la imagen.

Luego puse mis estadísticas de la cuenta con ayuda de una página web, la cual pone las estadísticas gráficamente bonitas y uno puede poner la cuenta de uno para que sí sean las estadísticas de la cuenta de uno, pero para poder pasar eso a código, copié la imagen y le dije a una Inteligencia Artificial si me ayudaba a ponerlo como código HTML. Con eso tengo todas las estadísticas de GitHub en mi perfil gráficamente bonito.

Al final solo agregué las redes sociales que tengo con los iconos iguales a los que puse en los lenguajes de programación por la estética, también en una tabla horizontalmente con los nombres de las redes sociales abajo de cada icono, con la diferencia que si alguien toca los iconos, los redirige a mi cuenta de la red social que tocaron. Abajo de esto puse mi correo electrónico si alguien me quiere contactar por si acaso, y eso sería todo mi README.

### III. CONCLUSIONES

El uso de Sistemas de Control de Versiones (VSC) como Git y plataformas colaborativas como GitHub representa una herramienta fundamental para la gestión eficiente de archivos y proyectos. Estos sistemas permiten llevar un historial detallado de cambios, recuperar versiones anteriores, evitar la pérdida de información y facilitar el trabajo en equipo.

Como estudiante de Sistemas Digitales 3, dominar estas herramientas me permitirá gestionar de manera organizada el código y la documentación de mis prácticas y proyectos, coordinarme con mis compañeros en el desarrollo de sistemas y asegurar que cualquier cambio realizado sea seguro y reversible. También me prepara para enfrentar de forma profesional proyectos reales en el campo de la ingeniería electrónica y la programación.

El repositorio README en Github sirve para uno identificarse como programador, también para colocar sus estadísticas de lo que sabe y de cuanto usa Github en su cuenta o también para que lo puedan contactar en diferentes redes sociales, eso hace una mejor comunicación con los demás programadores de diferentes naciones.

### REFERENCES

- [1] S. Chacon and B. Straub, *Pro Git: Getting Started – About Version Control*, Git-scm.com. [Online]. Available: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>. [Accessed: Aug. 7, 2025].
- [2] GitHub Docs, *Acerca de GitHub y Git*. [Online]. Available: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>. [Accessed: Aug. 7, 2025].
- [3] A. C. S. A., “¿Qué es GitHub y qué le ofrece a los desarrolladores?,” Xataka. [Online]. Available: <https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>. [Accessed: Aug. 7, 2025].
- [4] Vercel, “GitHub Stats Generator,” [Online]. Available: <https://gh-stats-gen.vercel.app/>. [Accessed: Aug. 7, 2025].
- [5] OpenAI, “ChatGPT,” [Online]. Available: <https://chat.openai.com/chat>. [Accessed: Aug. 7, 2025].