

1. Versionar tu código.

Esto quiere decir guardar en determinado momento los cambios efectuados a un archivo o conjunto de archivos, con la oportunidad que tener **acceso a ese historial de cambios**, ya sea para regresar a una de esas versiones o para hacer comparaciones entre ellas. En este punto hay que destacar las ventajas frente a herramientas como Subversion y Perforce; que son usadas para este mismo fin, la principal es que no necesitarías montar, mantener, hacer copias de seguridad, etcétera a un repositorio o servidor para proteger el código que escribes, en este caso Github es el plataforma que cumple esta función. Lo único que necesitas es instalar Git en tu equipo, crear tu cuenta en Github y luego de realizar algunas configuraciones sencillas y ejecutar los comandos adecuados podrás enviar tu código a Github. Es importante mencionar que la visibilidad de los proyectos será público o privado según el tipo de cuenta que adquieras.

2. Aprender y experimentar

Como muchos desarrolladores tienen de forma pública sus proyectos en Github **es posible acceder a su código, leerlo, estudiarlo y aprender de él, e incluso podrías hacer cambios y experimentar sin afectar el código original**, a esto último se le

como como: hacer [fork] (<https://help.github.com/articles/fork-a-repo/>)

3. Contribuir

Si luego de copiar un proyecto (hacer Fork) haces ajustes que arreglan bugs o introducen una nueva funcionalidad, puedes proponerle al dueño del proyecto que integre tus cambios en su código. Eso lo puedes hacer enviando un [pull request](#) con todas tus modificaciones o novedades. Mediante un pull request el administrador del repositorio original tendrá la posibilidad de revisar, testear o escribir comentarios sobre alguna líneas de tu propuesta, y finalmente si considera interesante tu contribución podrá realizar la fusión de esta con el código original (a esto se le conoce como hacer [merge](#) o en caso contrario rechazar los cambios propuestos.

4. Trabajo en equipo

Github es **el lugar perfecto para trabajar conjuntamente en una idea** con un amigo o colega, después de crear tu proyecto podrás invitar a otros usuarios como colaboradores, permitiendo así que estos lean y escriban directamente sobre tu repositorio.

5. Estar informado(a)

Github tiene un sistema de notificaciones con el cual puedes estar al tanto de las actividades alrededor de un repositorio en el cual estas participando o uno en el cual estés interesado. La notificaciones de los

proyectos a los que perteneces o eres dueño, están habilitadas de forma automática, y para aquellos en los cuales no eres participante pero estas interesado es necesario activarlas haciendo [watch]([https://help.github.com/articles](https://help.github.com/articles/watching-repositories/) la /watching-repositories/) sobre el repositorio en cuestión. Algunas de novedades que se notifican son: la creación de un pull request o el reporte de un bug, un mención de tu usuario en una conversación o comentario, etc. Estas notificaciones se pueden o consultar a través de correo electrónico o directamente en Github.

.

7. Mostrar tus conocimientos

Con Github puedes mostrar tus habilidades como desarrollador(a), puesto que es el código escrito en los archivos, donde reposa el resultado del proceso del desarrollo de software. Al compartir tu cuenta de Github con tu potencial empleador o cliente, este podrá ver la calidad del código que escribes a través de los proyectos públicos que estén en tu cuenta. Todos los proyectos que se escriben para ejecutar una idea, aprender un nuevo lenguaje o tecnología son válidos al

momento de exhibir tus conocimientos, así que no dudes en publicarlo en tu cuenta. Como complemento a lo anterior, con [Github Pages](#) puedes crear incluso una página como [esta](#) que te sirva como portafolio, en la cual puedes escribir sobre ti, los conocimientos que posees o poner enlaces de los proyectos en los que has participado.

8. Registro de incidencias

Cada proyecto creado en Github incluye un sistema de seguimiento de problemas, del estilo sistema de tickets, este permite a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) **abrir un ticket** escribiendo en este los detalles un problema que tenga con tu software o una sugerencia sobre una función que le gustaría que fuera implementada.

9. Compatibilidad

Github es una plataforma web, por tanto es independiente del sistema operativo que utilices, y además Git que es la herramienta que si requiere instalación es compatible con todos los sistemas; Linux, OSX y Windows.

10. Precio

Github, es **completamente gratis e ilimitado para proyectos públicos**, es decir que todos podrán ver el código que estos contienen (aunque tu siempre tendrás el control sobre quien subirá cambios), sin embargo si deseas puedes tener proyectos privados adquiriendo uno de

planes que ofrece, los cuales van desde 7 a 50 dólares mensuales, permitiendo crear 5 y 50 repositorios privados respectivamente.