

Introducción Informática

Clase 3 - Git

Consigna:

Resumen de la clase 3

¿Qué es git?

Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados). Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

¿Por qué usar un sistema de control de versiones?

Usar un sistema de control de versiones (VSC), permite revertir archivos y proyectos enteros a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más.

¿Qué es un repositorio?

Es el lugar donde se irán almacenando los archivos de nuestro proyecto. En GitHub podemos tener la cantidad de proyectos que queramos, en donde a cada proyecto le corresponderá un repositorio.

Tipos de repositorios

Los repositorios que se alojan en GitHub los llamamos repositorios remotos, mientras que a los que se alojan en nuestra PC los llamamos repositorios locales. Es necesario crear un vínculo entre ambos para poder mantener actualizados los archivos locales que están conectados a ese repositorio en la nube.

¿Qué significa que un archivo está en seguimiento?

Cuando enviamos un archivo al repositorio, estamos diciéndole a Git que queremos hacerle un seguimiento al mismo a través del tiempo.

Es decir, necesitamos que se guarde el estado actual de ese archivo para que cada vez que hagamos un cambio nuevo y lo enviamos, podamos comparar estados y ver cómo estaba en determinado momento. Seguirlo a lo largo del proyecto.

Palabras clave

A modo de resumen, repasamos las palabras claves:

git init // crea el repositorio

git config user.name "nombreUsuario" // agrega nuestra identidad

git config user.email "email" // agrega nuestro email

git remote add origin https://... // apunta al repositorio remoto

git add . // agrega todos los archivos

git status // seguimiento del estado de los archivos

git commit -m "texto descriptivo // son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor. Los commits van a ser nuestro historial de cambios que se fueron haciendo en el proyecto.

git push origin <<main/master/branch>> // envía los cambios al repositorio remoto

git pull origin <<main/master/branch>> // descarga los cambios que existen en el repositorio remoto

git clone linkDelRepo // clonamos el repositorio en nuestra máquina