Semana 3 módulo 2

Visión General de Git/GitHub

Un sistema de control de versiones te permite hacer un seguimiento de los cambios a tus documentos.

Esto hace mucho mas fácil la recuperación de versiones antiguas de tu documento en caso de cometer un error

y hace que la colaboración con otros sea mucho más fácil.

Aquí tenemos un ejemplo para ilustrar cómo funciona el control de versiones.

Vamos a decir que tienes una lista de compras y quieres que tus compañeros de habitación confirmen las cosas

que necesitas y agregar otros artículos adicionales.

Sin el control de versiones, tienes un gran desastre que limpiar antes de ir a realizar las compras.

Con el control de versiones, tu sabes EXACTAMENTE lo que necesitas después de que cada uno haya contribuido con sus ideas.

Reproduce el video desde :1:9 y sigue la transcripción1:09

Git es un software libre y de código abierto distribuido bajo la GNU General Public License.

Git es un sistema de control de versiones distribuido, lo que quiere decir que los usuarios de cualquier parte del mundo

pueden tener una copia de tu proyecto en su propia computadora, cuando ellos hayan hecho cambios,

pueden sincronizar su versión con un servidor remoto para compartirlo contigo.

Git no es el único sistema de control de versiones que existe allá fuera, pero el aspecto distribuido es una

de las principales razones por las que se ha convertido en uno de los sistemas de control de versiones más común disponible.

Los sistemas de control de versiones son ampliamente utilizados para cosas que implican código, pero igualmente puedes

controlar imágenes, documentos y cualquier tipo de archivo.

Puedes utilizar Git sin ninguna interfaz web usando tu interfaz de línea de comandos, pero

GitHub es uno de los servicios de alojamiento web más populares para los repositorios de Git.

Otros incluyen GitLab, BitBucket y Beanstalk.

Existen algunos términos que necesitarás conocer antes de poder comenzar.

El protocolo SSH es un método para acceso remoto seguro de una computadora a otra.

Un repositorio contiene las carpetas de tu proyecto que están configuradas para el control de versiones.

Una bifurcación es una copia de un repositorio.

Una pull request es la manera en que solicitas que alguien revise y apruebe tus cambios

antes de que se conviertan en la versión final.

Un directorio de trabajo contiene los archivos y subdirectorios de tu computadora los cuales están asociados

con un repositorio Git.

Hay algunos comandos básicos de Git los cuales siempre usarás.

Cuando comiences con un nuevo repositorio, solamente necesitarás crearlo una vez: bien sea localmente

y luego hacer push a GitHub o clonando un repositorio ya existente utilizando el comando “git init”.

Reproduce el video desde :3:3 y sigue la transcripción3:03

“git add” mueve los cambios desde el directorio de trabajo a la staging area.

“git status”, permite ver el estado de tu directorio de trabajo y la snapshot del stage

de tus cambios.

“git commit”, toma una captura instantánea del stage de los cambios y los confirma en el proyecto.

“git reset”, deshace los cambios que hayas hecho en los archivos de tu directorio de trabajo.

“git log”, permite navegar por los cambios previos de un proyecto.

“git branch”, permite crear un entorno aislado dentro de tu repositorio para hacer los cambios.

“git checkout”, permite ver y cambiar las branches existentes.

“git merge” permite volver a unir todo nuevamente.

Para aprender a usar Git efectivamente y comenzar a colaborar con los científicos de datos alrededor

del mundo, necesitarás aprender los comandos esenciales.

Por suerte para nosotros, GitHub tiene increíbles recursos disponibles para ayudarte a comenzar.

Visita try.github.io para descargar las hojas con los distintos trucos y repasar los tutoriales.

En los siguientes módulos, te daremos un curso intensivo acerca de cómo configurar tu entorno local

y como comenzar un proyecto.