

Mochila del viajero

Clase 4

1. ¿Quién inventó el sistema de control de versión Git y por qué?

El creador de Git fue Linus Torvalds, el creador de Linux. Lo hizo ya que quería un sistema de control de versiones que fuera de código abierto para trabajar con los equipos de desarrollo de Linux y respondiera a sus necesidades.

2. ¿A quién pertenece actualmente Github y por qué?

El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares. Esto se dio dado que la compañía contaba con una nueva CEO, Satya Nadella, que cambió el rumbo de la empresa hacia un mindset mucho más cercano al código Open Source y que trata de ganar seguidores de esa gran comunidad.

3. ¿Hay otra forma que no sea la terminal para trabajar con Github?

Sí, se puede trabajar mediante interfaces gráficas (GUIs) que permiten que usuarios sin experiencia con las interfaces de líneas de comando puedan hacer uso de esta herramienta sin problemas.

RESUMEN DE LA CLASE

Github

Aplicación estable, segura, rápida y eficiente. Es un lugar en la nube donde podés alojar archivos de tu proyecto de manera gratuita. Una vez creada la cuenta podemos comenzar a crear repositorios, donde se irán almacenando archivos del proyecto y a través del cual se puede hacer seguimiento del mismo. A cada proyecto le corresponde un repositorio. Los repositorios que viven en Github son repositorios remotos. Cada persona del equipo va a tener una copia del repositorio en su computadora, que son los repositorios locales.

Subiendo archivos

Usamos el comando **git push origin master**. Este repositorio le indica a Git que suba los archivos del repositorio local al remoto, en la nube. "Origin" se refiere a que es el remoto y "master" o "main" se refiere a que es la rama principal del proyecto. Tiene que estar todo commiteado.

Ramas

Copia alternativa del proyecto. Es una línea paralela en las que puedes agregar nuevas funcionalidades sin tener que modificar la línea original del proyecto. Es como una versión 2 del proyecto y si nos gusta la podemos mergear.

Bajando archivos

Usamos el comando **git clone URL_exacta** (donde se encuentran los archivos) para crear una copia exacta de todos los archivos existentes en el repositorio remoto. Se hace solo una vez en cada computadora.

Cuando lo que se quiere es actualizar los archivos pero ya se hizo antes el git clone debemos usar el comando **git pull origin master** (o el nombre de la rama). Siempre haremos un git pull antes de hacer un push para asegurarnos de no pisar nada cuando subamos los cambios.

Resolviendo conflictos

Si subimos versiones diferentes al mismo tiempo, Git hace un merge de ambas y las compatibiliza. Pero si estuvimos trabajando en el mismo archivo en el mismo lugar, Git no siempre sabe solucionar este problema. En este caso, Git nos indicará si tenemos conflictos y cómo solucionarlos. Pero en caso de que eso suceda, debemos corregirlo manualmente y hacer un nuevo commit. Por esto siempre debemos hacer un pull antes de un push. Si en este pull nos tira el error de que no puede fusionar ciertas líneas, las debemos trabajar manualmente. Entre los signos menor <<<< y los signos igual ==== vemos lo que tenemos en nuestro archivo, y debajo, después de los signos igual y antes de los signos mayor >>>> veremos lo que nos trae desde github. Luego de hacer la fusión manual debo hacer un nuevo add y un nuevo commit y finalmente un push.