



Gestor de Versiones

Problema del día

Tenemos a dos personas que están trabajando en un mismo archivo.

Persona A, está trabajando en la estructura del menú de navegación.


Persona B, está trabajando en la estructura Base del Footer.

Su jefe, les indica que deben mostrar en 5 minutos el avance al cliente en un solo proyecto.



¿Qué deberían hacer para
presentar el proyecto al cliente?

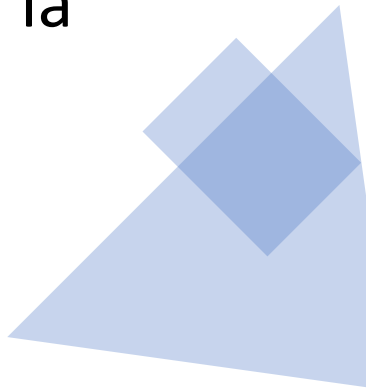





Entre ellos dos, deberán ponerse de acuerdo para que todos los cambios de un proyecto pasen al otro.

En este caso Persona B, debe transferir su contenido al archivo de Persona A.

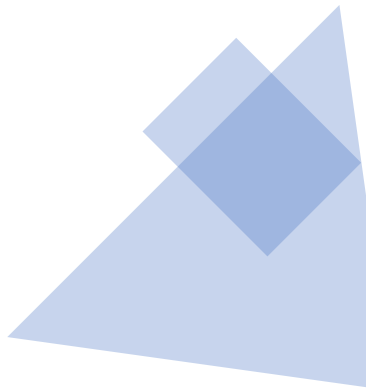
Por último, Persona A deberá pasarle una copia del nuevo proyecto unificado, el cual será la nueva base de trabajo.





Para este ejemplo, la solución era bastante simple, pero no exenta de posibles errores involuntarios al momento de la unificación.

Pero ¿qué pasaría si en vez de 2 personas son un equipo de trabajo de 10?, ¿y no es solamente un archivo sino 20 archivos por persona?





Es necesario automatizar el
proceso





GIT

Pensando en la eficiencia y la confiabilidad del mantenimiento de aplicaciones, cuando éstas tienen un gran número de archivos de código fuente, se recomienda encarecidamente el uso en equipos de trabajo.

Flujo de Trabajo

1. Se crea un Repositorio. (Punto en donde se encontrará el proyecto principal).
2. Se enlaza una carpeta local del equipo. (En ella trabajaremos escribiendo nuestro código).
3. Cuando terminemos de escribir nuestra funcionalidad, se deberá subir el cambio al repositorio.
4. Cada cierto tiempo puedo consultar por cambios hechos por mis compañeros para unificarlo a mi versión local.

La unificación en la mayoría de los casos es de forma automática. Pero habrá ocasiones en que GIT no sabrá como unificar, por lo que nos pedirá ayuda.

Gestionando y unificando

Recibiendo
cambios de
mis
compañeros



Enviando mis
nuevos códigos



Recibiendo cambios
de mis compañeros



Enviando mis
nuevos códigos.



`git init`

`git remote add origin
<dir repo>`

Realizar cambios en tu
editor favorito



Una sola vez por proyecto

Sube mis cambios que fueron
capturados por los commit.

`git push origin`

Agregar cambios
`git add .`



Obtengo los cambios de mis
compañeros.

`git pull origin`

Captura los cambios efectuados en el
proyecto local

`git commit -m "Cambios en
archivo A"`

