



ENTORNO DE TRABAJO Y CONTROL DE VERSIONES

Git: Evitando y solucionando problemas del
día a día

OBJETIVOS

- Enfrentar y resolver conflictos
- Ante errores, no entrar en pánico



REPASO DE LO VISTO HASTA AHORA

Recordemos lo visto hasta el momento...

- Sistema de control de versiones
- GitHub
- Repositorios
- Comandos básicos de Git
- Áreas o estados en Git
- Ramas
- GitFlow
- Ignorando archivos
- Lenguaje Markdown



GITHUB PAGES

GitHub Pages

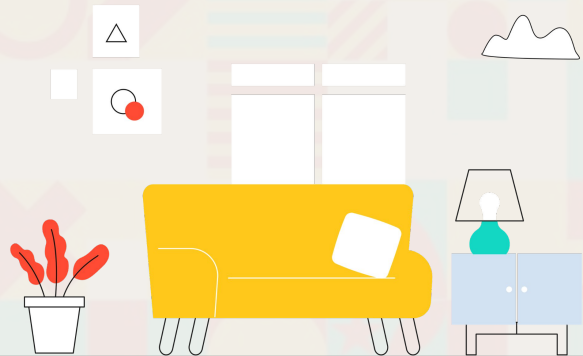
¿Sabías que puedes publicar tu proyecto web en pocos clicks?

GitHub Pages es un servicio de GitHub para hostear páginas estáticas, así que tu portfolio es el ejemplo perfecto con el que puedes iniciar.

Recuerda que al usar el plan gratuito de GitHub tu repositorio debe ser público.

Ahora, pasemos a hacerlo todos juntos!

Aquí te dejo también el link a la guía oficial: [github pages](https://docs.github.com/es/pages)

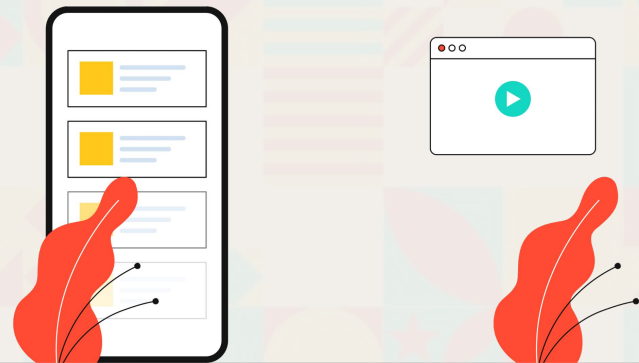


CONFLICTOS ERRORES Y MÁS

Git Amend

Si ya has estado trabajando con git puede que te haya pasado alguna vez que olvidaste de agregar cambio a tu commit, una opción es borrar el commit pero no siempre es la mejor opción.

Considera esto, si los cambios en el commit y el mensaje en sí son todos correctos pero te ha faltado agregar uno o más cambios lo mejor es utilizar `git commit --amend` para que el cambio que falta se incorpore al commit ya realizado.



Git Reset

Existen situaciones donde el commit hecho es sencillamente incorrecto y debes deshacer los cambios. Bueno, lo que debes hacer es lo siguiente:

- No hagas ningún otro cambio.
- Si quieres borrar el commit y quedarte con los cambios: `git reset --soft HEAD~1`
- Si quieres borrar el commit junto con los cambios: `git reset --hard HEAD~1`
- Si quieres retroceder más de un solo commit utiliza el hash:
`git reset --hard/--soft <hash del commit>`



Resolución de conflictos

Como ya sabemos, git nos da la libertad de que cada desarrollador tenga una copia del repositorio remoto y realizar modificaciones para luego llevarlas al repositorio remoto y que los demás actualicen los cambios.

A raíz de esto es que surge una situación muy común que es “conflicto” al juntar mis cambios con los cambios de otros desarrolladores.

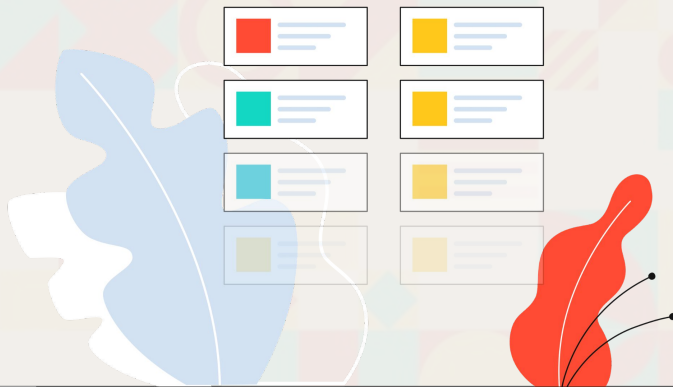
Entonces siempre que actualice tendré conflictos? No, esto sucede comúnmente cuando 2 desarrolladores modifican la misma línea del mismo archivo o si un desarrollador borró un archivo mientras otro realiza modificaciones en él.



Git Stash

Una situación que a veces se puede hacer común es que debas parar los cambios que estás haciendo porque otra tarea ha tomado prioridad. Esto obliga a que todos tus cambios hasta el momento los dejes en pausa y que no afecten de ninguna forma con la nueva tarea, para esto tienes las siguientes opciones:

- Manualmente deshaces todos los cambios e inicias con la nueva tarea. ¿En serio?
- Haces un commit sin importar si los cambios están completos o funcionan. Por favor no!
- Pausar los cambios con git stash.



Git Stash - Comandos

- Agregar cambios: `git stash push -m "mensaje"`
- Aplicar cambios: `git stash apply <index>`
- Mostrar lista: `git stash list`
- Aplicar cambios y borrar de la lista: `git stash pop <index>`
- Borrar toda la lista: `git stash clear`



Git Clean

Si has estado creando nuevos archivos que luego tienes que borrar, en lugar de revertir los cambios uno a uno, te sugiero que uses

```
git clean
```

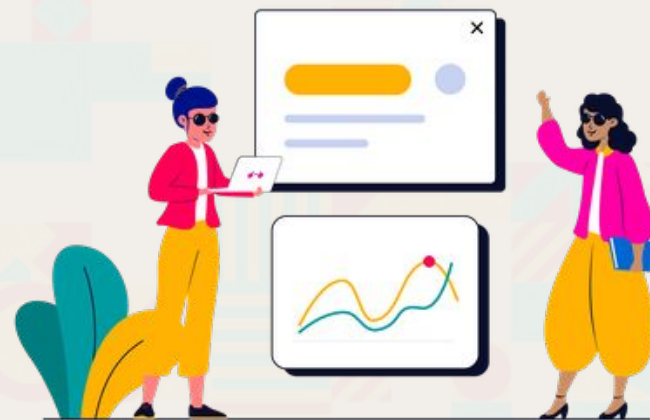
para limpiar tu área de trabajo e iniciar de cero. Recuerda que solo aplica a archivos sin seguimiento por parte de git.



Ahora vayamos a nuestras máquinas y practiquemos!

TAREAS

- Si te quedaron dudas respecto al procedimiento te recomendamos ver la grabación de la clase.
- Recuerda que también tienes las clases de consulta para dudas puntuales.
- Estos comandos tienen muchas más opciones, te recomiendo ver la documentación de git ([link](#))





Gobierno de JUJUY
Ministerio de Desarrollo
Económico y Producción
*Dirección Provincial de Servicios
Basados en el Conocimiento*



FACULTAD DE
INGENIERIA
UNIVERSIDAD NACIONAL DE JUJUY



INCLUIT