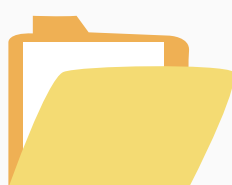


Domina el control de versiones y eleva tus proyectos.

01



## git init

Git init crea el repositorio local. Con este comando podremos hacer que git empiece a vigilar la carpeta de nuestro proyecto.

**Implementación:** C:\usuarios\mi-proyecto> git init

## git add

Área de espera. Aquí seleccionamos cuáles archivos modificados son los que se van a guardar.

**Implementación:** C:\usuarios\mi-proyecto> git add index.html

02



03



## git commit

Con este comando guardamos. Se genera copia y si algo pasa en el futuro, podrás regresar a ese punto exacto.

**Implementación:** C:\usuarios\mi-proyecto> git commit -m "Se agregó la estructura de la página".

## git push

Subir cambios. Se suben los cambios a tu cuenta de GitHub, ahora el código está respaldado de forma segura y disponible para personas (si es público).

**Implementación:** C:\usuarios\mi-proyecto> git push origin main

04



05



## Tu portafolio!

¡Proyecto publicado! Al subir tu código, no solo lo respaldas de forma segura: estás construyendo tu portafolio profesional en internet.

**Implementación:** Tu perfil web - <https://github.com/tu-usuario>

## Ramas - Trabajo en equipo

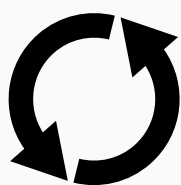
Para el trabajo en equipo existen las ramas (branches). Te permiten clonar el proyecto principal, añadir funciones de forma aislada y después fusionarlo con el proyecto final.

**Implementación:** C:\usuarios\mi-proyecto> git checkout -b mi-funcion

06



07



## git pull

Cuando trabajas en equipo, tus compañeros subirán cambios a GitHub. Con este comando descargas esas actualizaciones a tu computadora para tener la última versión del proyecto.

**Implementación:** C:\usuarios\mi-proyecto> git pull origin main