

Git Basico

Guía Básica de Git para Colaboración



Grupo Flores

1. Introducción

Este documento explica los conceptos más básicos para trabajar en un proyecto compartido usando Git.

Está enfocado en personas que nunca han usado Git o apenas comienzan.

Aquí aprenderás:

- Cómo descargar un proyecto
- Cómo guardar cambios
- Cómo subir y bajar cambios
- Qué significan los comandos más comunes

2. ¿Cómo funciona Git en equipo?

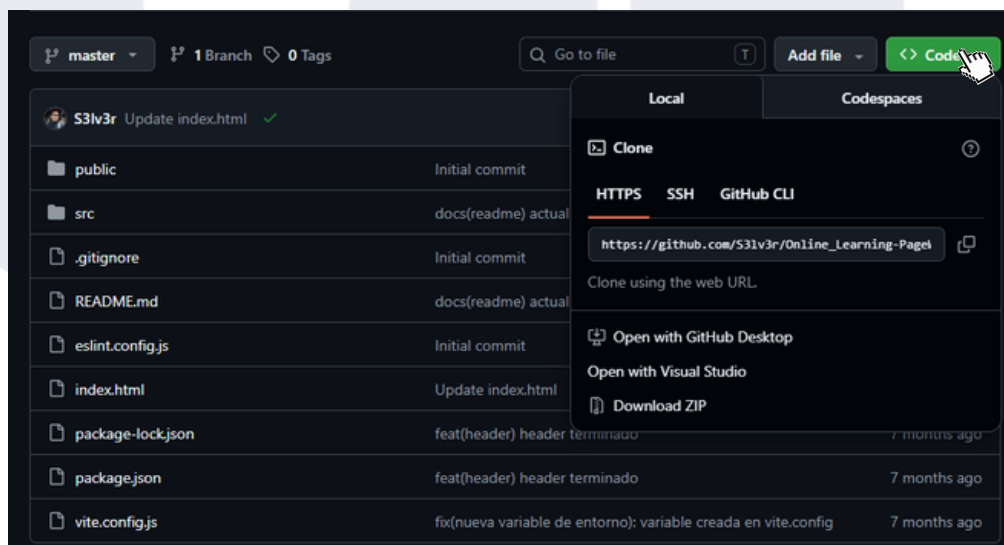
Cuando trabajas con otras personas:

- El proyecto está en internet (GitHub)
- Cada persona tiene una copia en su computadora
- Todos pueden modificar el proyecto
- Los cambios se sincronizan constantemente

3. Clonar un repositorio

Clonar significa descargar una copia completa del proyecto a tu computadora.

```
git clone https://github.com/usuario/proyecto.git
```



La url es diferente para cada repositorio, en la imagen se muestra como obtenerla

Después entra a la carpeta:

```
cd proyecto
```

4. ¿Qué es un commit?

Un commit es un guardado de cambios en el proyecto.

Git no guarda automáticamente, tú decides cuándo guardar.

```
git add .  
git commit -m "mensaje"
```

Esto crea un punto en el historial del proyecto.

5. ¿Qué hace git add?

Antes de hacer un commit, necesitas decirle a Git qué archivos quieres guardar.

Opción 1: Agregar todo

```
git add .
```

Agrega todos los archivos modificados.

Opción 2: Agregar un archivo específico

```
git add card.jsx
```

Solo agrega ese archivo.

6. ¿Qué es un push?

Push significa subir tus cambios a GitHub.

```
git push
```

Sin esto:

- Tus cambios solo están en tu computadora
- Nadie más los puede ver

7. ¿Qué es un pull?

Pull significa descargar cambios del proyecto en Github.

```
git pull
```

Esto actualiza tu proyecto con lo que otros han hecho.

8. ¿Qué es git stash?

git stash sirve para guardar cambios temporalmente sin hacer commit.

Se usa cuando:

- Estás trabajando
- Pero necesitas actualizar el proyecto
- No quieres hacer commit todavía

```
git stash
```

Esto guarda tus cambios y deja tu proyecto limpio.

9. ¿Qué es git stash pop?

Recupera los cambios que guardaste con stash:

```
git stash pop
```

Esto:

- Devuelve tus cambios
- Elimina ese guardado temporal

Ejemplo de stash:

```
git stash  
git pull  
git stash pop
```

Flujo:

1. Guardas tus cambios temporalmente
2. Actualizas el proyecto
3. Recuperas lo que estabas haciendo

10. Orden correcto al trabajar

Antes de empezar:

```
git pull
```

Después de hacer cambios:

```
git add .  
git commit -m "mensaje claro"  
git push
```

11. Reglas básicas

- Siempre hacer git pull antes de trabajar
- Siempre hacer git push después
- Usar git add correctamente (no siempre .)
- Usar git stash si necesitas pausar cambios
- No asumir que eres el único trabajando

12. Conclusión

Si entiendes esto:

- **clone** → descargar proyecto
- **add** → seleccionar cambios
- **commit** → guardar cambios
- **push** → subir cambios
- **pull** → bajar cambios
- **origin** → repositorio en internet
- **stash** → guardar temporalmente

Ya tienes lo necesario para colaborar sin romper el proyecto.

Es importante señalar que esta información debe complementarse con el contenido del PDF titulado “Git Workflow”.