Subir una Aplicación Web a Git y GitHub

# 1. Inicializa el repositorio local con Git

Abre Git Bash dentro de la carpeta del proyecto y ejecuta:  
  
```bash  
git init  
```  
Esto crea el repositorio local.

# 2. Agrega todos los archivos al staging

```bash  
git add .  
```  
Esto agrega todos los archivos actuales al área de preparación.

# 3. Haz tu primer commit

```bash  
git commit -m "Primer commit del proyecto web"  
```  
Guarda una versión del estado actual del proyecto.

# 4. Crea un repositorio en GitHub

1. Ve a https://github.com  
2. Haz clic en "New repository"  
3. Ponle nombre al repositorio (ej. mi-aplicacion-web)  
4. NO marques la opción “Initialize with README”  
5. Copia la URL del repositorio (ej. https://github.com/usuario/mi-aplicacion-web.git)

# 5. Conecta tu repositorio local con GitHub

```bash  
git remote add origin https://github.com/usuario/mi-aplicacion-web.git  
```  
Esto enlaza tu repositorio local con el remoto en GitHub.

# 6. Sube los archivos a GitHub

```bash  
git push -u origin main  
```  
Si falla por el nombre de la rama, usa:  
  
```bash  
git push -u origin master  
```  
Esto sube los archivos a GitHub.

# Actualizaciones futuras

Para subir cambios nuevos:  
  
```bash  
git add .  
git commit -m "Descripción del cambio"  
git push  
```

Comandos Basicos de Git

|  |  |  |
| --- | --- | --- |
| ITEM | COMANDO | DESCRIPCIÓN |
| 1 | git init | Inicializa un nuevo repositorio Git en el directorio actual. |
| 2 | git add. | Añade todos los archivos al área de preparación (staging area). |
| 3 | git reset | Quita archivos del área de preparación. |
| 4 | Git commit | Guarda los cambios preparados en el historial del repositorio. |
| 5 | Git checkout -- | |  | | --- | |  |  |  | | --- | | Deshace cambios en el archivo modificado (lo deja como estaba). | |
| 6 | Git log | |  | | --- | |  |  |  | | --- | | Muestra el historial de commits. | |
| 7 | Git commit –amend | Modifica el último commit (por ejemplo, para corregir un mensaje). |
| 8 | Git checkout -b | Crea y cambia a una nueva rama. |
| 9 | Git checkout mastes | Cambia a la rama principal master. |
| 10 | Git branch | |  | | --- | |  |  |  | | --- | | Lista todas las ramas del repositorio. | |
| 11 | Git push | Envía los cambios al repositorio remoto (como GitHub). |
| 12 | Git commit -am | Añade y hace commit de archivos modificados en un solo paso. |
| 14 | pwd | Muestra el directorio actual (no es un comando Git, es de la terminal). |
| 15 | ls | Lista los archivos y carpetas del directorio actual (comando de la terminal). |

Ejemplos Básicos y Aplicados de Comandos Git

|  |  |  |
| --- | --- | --- |
| COMANDO | EJEMPLO COMPLETO | EXPLICACIÓN |
| git init | git init | Inicia un repositorio Git vacío en el directorio actual del proyecto. |
| git add . | git add . | Agrega **todos** los archivos del proyecto al staging area para ser preparados al commit. |
| git reset | git reset archivo.html | Quita archivo.html del área de preparación. |
| git commit | git commit -m "Añadida página de contacto" | Guarda los archivos añadidos con un mensaje explicando el cambio. |
| git checkout -- | git checkout -- index.html | Deshace los cambios locales en index.html y restaura su versión desde el último commit. |
| git log | git log --oneline --graph --decorate --all | Muestra el historial de commits de forma compacta y visual (muy útil). |
| git commit --amend | git commit --amend -m "Corrijo el mensaje del último commit" | Reemplaza el mensaje del último commit (si no has hecho push). |
| git checkout -b | git checkout -b desarrollo-header | Crea una nueva rama llamada desarrollo-header y se cambia a ella automáticamente. |
| git checkout master | git checkout master | Cambia a la rama master (rama principal por defecto en muchos repositorios). |
| git branch | git branch git branch -d vieja-rama | Muestra todas las ramas. El segundo ejemplo elimina una rama local llamada vieja-rama. |
| git push | git push origin desarrollo-header | Envía la rama desarrollo-header al repositorio remoto llamado origin. |
| git commit -am | git commit -am "Corrijo estilos CSS del home" | Añade y hace commit **solo de archivos modificados** (no nuevos) con un mensaje. |
| pwd | pwd | Muestra la ubicación actual en el sistema de archivos. Útil para saber en qué carpeta estás. |
| ls | ls -l | Lista archivos del directorio actual con detalles como permisos, tamaño y fecha de modificación. |