

GIT Y GITHUB

1. INSTALAR GIT EN WINDOWS

- Descarga el instalador de Git desde git-scm.com.
- Ejecuta el instalador y sigue las instrucciones. Asegúrate de seleccionar la opción "**Git Bash**" durante la instalación, ya que te permitirá usar una terminal similar a la de Linux/macOS.
- Una vez instalado, abre **Git Bash** desde el menú de inicio.

2. CONFIGURAR GIT

- Abre **Git Bash** y configura tu nombre y correo electrónico:

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
```

- Verifica la configuración:

```
git config --list
```

3. Inicializar un Repositorio Git

- Navega a la carpeta de tu proyecto usando el comando cd. Por ejemplo:

```
cd /c/ruta/de/tu/proyecto
```

(En Windows, las unidades de disco se representan como /c/ para C:\, /d/ para D:\, etc.).

- Inicializa el repositorio:

```
git init
```

4. Agregar Archivos al Repositorio

- Agrega los archivos al área de preparación:

```
git add .
```

- Verifica el estado:

```
git status
```

5. Hacer tu Primer Commit

- Guarda los cambios con un mensaje descriptivo:

```
git commit -m "Primer commit: estructura inicial del proyecto"
```

6. Crear un Repositorio Remoto

- Si usas GitHub, GitLab o Bitbucket, crea un repositorio en la plataforma.
- Conecta tu repositorio local con el remoto:

```
git remote add origin https://github.com/tu-usuario/tu-repositorio.git
```

- Sube tus cambios:

```
git push -u origin main
```

(Nota: En algunos casos, la rama principal puede llamarse master en lugar de main).

7. Trabajar con Ramas (Branches)

- Crea una nueva rama:

```
git branch nombre-de-la-rama
```

- Cambia a la nueva rama:

```
git checkout nombre-de-la-rama
```

- O usa el atajo:

```
git checkout -b nombre-de-la-rama
```

- Fusiona la rama con main cuando termines:

```
git checkout main  
git merge nombre-de-la-rama
```

8. Actualizar tu Repositorio Local

- Si trabajas en equipo, actualiza tu repositorio local con los cambios remotos:

```
git pull origin main
```

9. Ignorar Archivos (.gitignore)

- Crea un archivo .gitignore en la raíz de tu proyecto. En Git Bash, usa:

```
touch .gitignore
```

- Edita el archivo .gitignore con un editor de texto (como Notepad o Visual Studio Code) y agrega los archivos o carpetas que desees ignorar:

```
node_modules/  
.env  
*.log
```

10. Explorar el Historial

- Revisa el historial de commits:

```
git log
```

- Para salir del historial, presiona q.
-

Notas adicionales para Windows:

1. **Git Bash vs Command Prompt (CMD) o PowerShell:**

- Puedes usar Git en **Command Prompt** o **PowerShell**, pero **Git Bash** es más recomendable porque tiene un comportamiento más similar al de Linux/macOS.
- Si prefieres usar **PowerShell**, los comandos son los mismos, pero la navegación de carpetas sigue la sintaxis de Windows (por ejemplo, `cd C:\ruta\de\tu\proyecto`).

2. **Rutas de archivos:**

- En Git Bash, las rutas de archivos usan barras / en lugar de barras invertidas \. Por ejemplo:

```
cd /c/Users/TuUsuario/Documents/proyecto
```

3. **Editores de texto:**

- Si necesitas editar archivos (como `.gitignore`), puedes usar cualquier editor de texto, como Notepad, Notepad++, Visual Studio Code, etc.

Si ya tienes la carpeta y los archivos de tu proyecto web creados, el proceso es muy similar. Solo necesitas inicializar Git en esa carpeta y agregar los archivos existentes al repositorio. Aquí te explico cómo hacerlo paso a paso:

1. Abre Git Bash y navega a tu carpeta

- Abre **Git Bash** desde el menú de inicio.
- Usa el comando `cd` para navegar a la carpeta de tu proyecto. Por ejemplo:

```
cd /c/ruta/de/tu/proyecto
```

(Recuerda que en Git Bash las rutas usan `/` en lugar de `\`).

2. Inicializa un repositorio Git

- Una vez dentro de la carpeta de tu proyecto, inicializa Git:

```
git init
```

- Esto creará una carpeta oculta `.git` en tu proyecto, que almacena toda la información de versiones.
-

3. Agrega los archivos existentes al repositorio

- Agrega todos los archivos de tu proyecto al área de preparación (staging area):

```
git add .
```

(El `.` agrega todos los archivos en la carpeta actual).

- Si solo quieres agregar archivos específicos, puedes usar:

```
git add nombre_archivo
```

4. Verifica el estado de los archivos

- Usa el siguiente comando para ver qué archivos están listos para ser confirmados (committed):

```
git status
```

- Verás una lista de archivos en verde (staged) que están listos para el commit.
-

5. Haz tu primer commit

- Guarda los cambios en el repositorio con un mensaje descriptivo:

```
git commit -m "Primer commit: agregando archivos existentes del proyecto"
```

6. Conectar con un repositorio remoto (opcional)

- Si quieres subir tu proyecto a un servicio como GitHub, GitLab o Bitbucket:

1. Crea un repositorio en la plataforma elegida.
2. Conecta tu repositorio local con el remoto:

```
git remote add origin https://github.com/tu-usuario/tu-repositorio.git
```

3. Sube tus cambios al repositorio remoto:

```
git push -u origin main
```

(Nota: Si tu rama principal se llama master en lugar de main, usa master).

7. Ignorar archivos no deseados (.gitignore)

- Si tienes archivos o carpetas que no quieres incluir en el repositorio (por ejemplo, node_modules, archivos de configuración local, etc.), crea un archivo .gitignore:

```
touch .gitignore
```

- Edita el archivo .gitignore con un editor de texto y agrega los nombres de los archivos o carpetas que deseas ignorar. Por ejemplo:

```
node_modules/  
.env  
*.log
```

- Luego, agrega y confirma el archivo .gitignore:

```
git add .gitignore  
git commit -m "Agregar archivo .gitignore"
```

8. Trabajar con ramas (opcional)

- Si quieres trabajar en una nueva funcionalidad o corrección sin afectar la rama principal (main o master), crea una nueva rama:

```
git checkout -b nombre-de-la-rama
```

- Cuando termines, fusiona la rama con la rama principal:

```
git checkout main  
git merge nombre-de-la-rama
```

9. Actualizar tu repositorio local (si trabajas en equipo)

- Si trabajas en equipo, asegúrate de mantener tu repositorio local actualizado con los cambios remotos:

```
git pull origin main
```

10. Explorar el historial

- Revisa el historial de commits para ver los cambios realizados:

```
git log
```

- Para salir del historial, presiona q.
-

Resumen

Si ya tienes una carpeta con archivos, solo necesitas:

1. Inicializar Git en la carpeta (`git init`).
2. Agregar los archivos existentes (`git add .`).
3. Hacer un commit (`git commit -m "Mensaje"`).
4. (Opcional) Conectar con un repositorio remoto y subir los cambios (`git remote add origin` y `git push`).