**Conceptos básicos de Git**

**CONTENIDO**

[**¿Qué es Git? 3**](#_heading=h.tnnukpegwk5a)

[**Comandos Básicos de Git 3**](#_heading=h.x8idbvdqep5k)

[**¿Qué es GitHub? 4**](#_heading=h.d2u5mu5j0znr)

[**Funciones Principales de GitHub 4**](#_heading=h.bk1fur7dp7cb)

[**Flujo de Trabajo Básico en GitHub 4**](#_heading=h.2jz3jxccjvus)

[**Acceso al Terminal en Visual Studio Code 4**](#_heading=h.2dn4y2wcgvc)

[**Uso de Git en Visual Studio Code instalado en equipo con Windows 10 4**](#_heading=h.yuxo54ndji6)

### 

### **¿Qué es Git?**

Git es un sistema de control de versiones distribuido diseñado para manejar desde proyectos pequeños a muy grandes con velocidad y eficiencia. Permite a los desarrolladores guardar "instantáneas" del código en diferentes puntos del tiempo, que luego pueden ser accedidas o restauradas.

### **Comandos Básicos de Git**

Git tiene una amplia gama de comandos que permiten la gestión efectiva del código fuente de un proyecto:

**git init**: Inicializa un nuevo repositorio de Git en el directorio actual.  
bash  
git init

**git clone [url]**: Clona un repositorio remoto a tu máquina local.  
bash  
git clone https://github.com/ejemplo/repo.git

**git add [archivo]**: Añade un archivo al área de preparación ("staging area"), preparándolo para ser incluido en la próxima "commit".  
bash  
git add archivo.txt

**git commit -m "[mensaje]"**: Crea una nueva "commit" con los cambios añadidos y le asigna un mensaje descriptivo.  
bash  
git commit -m "Añadido archivo.txt al proyecto"

**git push**: Empuja las "commits" desde tu repositorio local a un repositorio remoto.  
bash  
git push origin master

**git pull**: Actualiza tu rama local con la versión más reciente de la rama remota.  
bash  
git pull origin master

**git status**: Muestra el estado del repositorio local, incluyendo cambios que no han sido preparados para "commit".  
bash  
git status

**git branch**: Gestiona las ramas de tu repositorio.  
bash  
git branch nueva-rama

**git checkout [nombre-rama]**: Cambia a otra rama existente o crea una nueva y cambia a ella.  
bash  
git checkout -b nueva-rama

### **¿Qué es GitHub?**

GitHub es una plataforma de hospedaje de código para control de versiones y colaboración. Permite a múltiples personas trabajar juntas en proyectos desde cualquier parte del mundo. GitHub se basa en Git y lo extiende con características como la interfaz web, control de acceso, herramientas de colaboración, entre otros.

### **Funciones Principales de GitHub**

* **Repositorios**: Almacenan el código fuente del proyecto y el historial de todas las "commits".
* **Forks**: Copias de un repositorio que pertenecen a otra cuenta, permitiendo a los usuarios proponer cambios a un proyecto externo sin afectar al original.
* **Pull Requests**: Solicitudes enviadas por los usuarios para que los mantenedores de un repositorio consideren incorporar sus cambios.
* **Issues**: Sistema de seguimiento de problemas que permite a los usuarios reportar bugs o solicitar nuevas características.
* **GitHub Actions**: Permite automatizar flujos de trabajo directamente desde GitHub.

### **Flujo de Trabajo Básico en GitHub**

El flujo de trabajo típico cuando se usa GitHub incluye:

1. **Fork** (copia independiente) de un repositorio.
2. **Clonar** el fork a tu máquina local.
3. **Crear una rama** para cada nueva característica o corrección.
4. Realizar **cambios** y **commit** a esa rama.
5. Empujar la rama a GitHub.
6. Abrir un **pull request** desde la nueva rama del fork hacia el repositorio original.
7. **Revisión de código** y eventualmente **fusionar** la rama con el repositorio principal.

### **Acceso al Terminal en Visual Studio Code**

1. **Abrir VS Code**.
2. **Abrir el terminal**: Puedes hacerlo navegando a Terminal en la barra de menú y luego seleccionando New Terminal, o simplemente utilizando el atajo de teclado que generalmente es `Ctrl+`` (control y la tecla de acento grave).

### **Uso de Git en Visual Studio Code instalado en equipo con Windows 10**

El uso de comandos de Git en Visual Studio Code aplica igualmente para Windows 10. Los comandos de Git son consistentes a través de diferentes sistemas operativos y terminales, incluyendo el terminal integrado en VS Code en Windows 10. Esto significa que puedes ejecutar los mismos comandos de Git en VS Code en Windows, macOS, o cualquier otro sistema operativo compatible, siempre y cuando Git esté instalado en tu sistema.