

Tutorial de GitHub

Aquí tienes un tutorial básico para aprender a usar GitHub. GitHub es una plataforma de control de versiones que utiliza **Git** como sistema de gestión de versiones. Te permite colaborar en proyectos, gestionar el código y realizar seguimiento de cambios.

1. **Crear una cuenta en GitHub**

Primero, necesitas una cuenta en GitHub:

1. Ve a github.com.
2. Regístrate y crea tu perfil.
3. Luego, confirma tu correo electrónico.

2. **Instalar Git**

Si no tienes Git instalado en tu computadora, sigue estos pasos:

- Para Windows:

1. Descarga Git desde git-scm.com y sigue las instrucciones del instalador.

- Para macOS:

1. Usa Homebrew para instalar Git:

```
``bash
brew install git
``
```

- Para Linux (Debian/Ubuntu):

1. Ejecuta:

```
``bash
sudo apt-get install git
``
```

3. **Configurar Git**

Una vez instalado Git, configúralo con tu nombre de usuario y correo electrónico, que asociará tus cambios en GitHub.

```
``bash
git config --global user.name "TuNombre"
git config --global user.email "TuCorreo@ejemplo.com"
``
```

4. **Crear un repositorio en GitHub**

1. Inicia sesión en GitHub.
2. Haz clic en el botón de **New repository**.

3. Asigna un nombre a tu repositorio y selecciona si será ****público**** o ****privado****.
4. Puedes agregar un archivo README para describir tu proyecto. Luego, haz clic en ****Create repository****.

5. ****Clonar el repositorio a tu máquina local****

Para trabajar en tu computadora, clona el repositorio de GitHub:

1. Ve a tu repositorio recién creado y copia el enlace HTTPS o SSH.
2. En la terminal (o Git Bash en Windows), ejecuta el siguiente comando:

```
``bash
git clone https://github.com/tuusuario/tu-repositorio.git
``
```

Este comando descargará el repositorio en una carpeta local.

6. ****Hacer cambios y subirlos a GitHub****

a) ****Agregar archivos****

Agrega un archivo en la carpeta del repositorio local. Por ejemplo, crea un archivo llamado ``index.html``.

b) ****Ver el estado del repositorio****

Puedes ver los archivos que has modificado o agregado con:

```
``bash
git status
``
```

c) ****Agregar archivos a la preparación (staging)****

Antes de subir los cambios a GitHub, primero debes agregarlos al área de preparación (staging) para indicar qué cambios quieres incluir:

```
``bash
git add index.html
``
```

d) ****Hacer un commit****

Un commit es un registro de los cambios que has realizado. Para crear un commit con un mensaje describiendo los cambios, ejecuta:

```
``bash
git commit -m "Añadí el archivo index.html"
```

```
'''
```

```
##### e) **Subir los cambios a GitHub (push)**
```

Finalmente, sube los cambios a GitHub con el siguiente comando:

```
'''bash
git push origin main
'''
```

Nota: Si estás trabajando en una rama diferente a `main`, reemplaza `main` por el nombre de tu rama.

```
### 7. **Hacer un seguimiento de cambios**
```

Git guarda un historial de todos los commits. Para ver el historial de commits en tu repositorio local:

```
'''bash
git log
'''
```

```
### 8. **Crear y trabajar con ramas (branches)**
```

Las ramas te permiten trabajar en diferentes versiones de tu proyecto sin afectar el código principal.

```
##### a) **Crear una nueva rama**
```

Para crear y cambiar a una nueva rama:

```
'''bash
git checkout -b nueva-rama
'''
```

```
##### b) **Subir los cambios de la nueva rama a GitHub**
```

Realiza los cambios, haz commit y súbelos a la nueva rama con:

```
'''bash
git push origin nueva-rama
'''
```

```
### 9. **Pull Request (Solicitar una fusión de cambios)**
```

Cuando termines de trabajar en una rama, puedes crear un **pull request** para solicitar que los cambios se fusionen en la rama principal:

1. Ve a tu repositorio en GitHub.
2. Haz clic en **Pull requests**.
3. Haz clic en **New pull request**, selecciona la rama y revisa los cambios.
4. Si todo está bien, solicita la fusión o pide a alguien que la revise.

10. **Actualizar el repositorio local**

Si otros contribuyen al mismo proyecto, debes mantener tu repositorio local actualizado. Para actualizarlo, usa:

```
``bash
git pull origin main
``
```

Esto descargará y aplicará los últimos cambios del repositorio remoto a tu copia local.

Este es un resumen básico de cómo empezar a usar GitHub. A medida que te familiarices con Git y GitHub, podrás explorar otras funciones avanzadas como las **acciones de GitHub**, el **manejo de conflictos** y la **gestión de colaboraciones**. ¡Espero que te sirva de ayuda!