

## GIT

Comandos Principales Git.....	1
git init.....	1
git clone.....	1
git add.....	1
git commit .....	1
git status.....	1
git push.....	1
git pull.....	1
git Branch.....	1
git checkout.....	1
git merge .....	1
COMANDOS DE CONFIGURACION .....	2
git config --global user.name "Tu Nombre" .....	2
git config --global user.email tu@email.com .....	2
git config --global alias.[alias] "[comando]" .....	2
git config --global color.ui auto.....	2
git config --global core.editor [editor] .....	2
git config --global core.autocrlf true   input .....	2
git config --global merge.tool [herramienta].....	2

## COMANDOS PRINCIPALES GIT

### git init

Inicializa un repositorio Git en un directorio, creando un nuevo repositorio vacío o reutilizando uno existente en esa carpeta.

### git clone

Clona un repositorio Git existente en un nuevo directorio. Se utiliza para obtener una copia local de un repositorio remoto.

### git add

Añade cambios al área de preparación. Puedes utilizar `git add .` para agregar todos los cambios, o `git add nombre_archivo` para agregar archivos específicos.

### git commit

Guarda los cambios confirmados en el repositorio. Utiliza `git commit -m "Mensaje del commit"` para realizar un commit con un mensaje.

### git status

Muestra el estado actual de los archivos en el directorio de trabajo y en el área de preparación. Indica los archivos modificados, agregados o eliminados.

### git push

Sube los commits locales al repositorio remoto. Por lo general, se usa como `git push origin nombre_rama` para enviar cambios a la rama especificada en el repositorio remoto llamado 'origin'.

### git pull

Descarga cambios del repositorio remoto a tu repositorio local. Fusiona los cambios de la rama remota en tu rama local actual.

### git Branch

Muestra las ramas presentes en el repositorio. También se puede usar para crear, eliminar o cambiar entre ramas.

### git checkout

Cambia entre ramas o restaura archivos. Se utiliza para cambiar entre ramas, crear nuevas ramas o restaurar archivos a un estado anterior.

### git merge

Fusiona cambios de una rama a otra. Permite combinar una rama con otra rama actual.

## COMANDOS DE CONFIGURACION

```
git config --global user.name "Tu Nombre"
```

Establece tu nombre de usuario para los commits. Reemplaza "Tu Nombre" con tu nombre real entre comillas.

```
git config --global user.email tu@email.com
```

Asigna tu dirección de correo electrónico para los commits. Cambia "tu@email.com" por tu dirección de email.

```
git config --global alias.[alias] "[comando]"
```

Crea alias para comandos Git. Sustituye [alias] por el nombre del alias que deseas y [comando] por el comando al que deseas asignarlo. Por ejemplo, `git config --global alias.co checkout` crea un alias "co" para el comando checkout.

```
git config --global color.ui auto
```

Habilita el coloreado de la salida en la línea de comandos para mejorar la legibilidad.

```
git config --global core.editor [editor]
```

Define el editor por defecto para los mensajes de commit. Reemplaza [editor] con el nombre de tu editor preferido, por ejemplo, "vim", "nano", "emacs" o "code" para Visual Studio Code.

```
git config --global core.autocrlf true | input
```

Configura la conversión automática de saltos de línea. true es para sistemas Windows y input para sistemas Unix-like.

```
git config --global merge.tool [herramienta]
```

Establece la herramienta de fusión que prefieres utilizar en caso de conflictos. Por ejemplo, "vimdiff", "meld", "kdiff3", entre otros.