

Apuntes de GIT

1. Introducción

1.1. Qué es un sistema de control de versiones?

Un sistema de control de versiones es un software que nos permite registrar los cambios que son realizados en un archivo o un conjunto de archivos a lo largo del tiempo, no es útil por el hecho de la facilidad de que varias personas trabajen en un mismo proyecto al mismo tiempo, tambien nos ayuda a que en caso de un error podamos volver a versiones pasadas funcionales de dicho proyecto.

1.2. Fundamentos de GIT

GIT se basa en los repositorios, un repositorio no es mas que el lugar donde se alojan las versiones de un proyecto junto con el log de cambios que se han hecho en ese projects. Dichos repositorios pueden ser tanto locales como remotos

- Locales.

Los repositorios locales son los que tenemos en nuestra computadora.

- Remoto.

Los repositorios remotos son los que se ubican en un servidor, los cuales permiten que varias personas vean nuestro proyecto, que realicen cambios en el y que esos cambios sean sincronizados.

1.3. Configuración inicial

Antes de comenzar a usar GIT, se debe realizar ciertas configuraciones

1.3.1. Correo electronico

Para indicarle a GIT nuestro correo electrónico debemos ejecutar este comando:

```
$ git config --global user.email "reemplaza esto por tu correo"
```

Por ejemplo:

```
$ git config --global user.email "jschavarria77@gmail.com"
```

1.3.2. Nombre

Para indicarle a GIT nuestro nombre debemos ejecutar el siguiente comando:

```
$ git config --global user.name "reemplaza esto por tu nombre completo"
```

Por ejemplo:

```
$ git config --global user.name "Jaime Sebastian Chavarria Fuertes"
```

1.4. Configurar el editor de código que abre GIT

GIT tiene configurado abrir por defecto el editor de texto Vim para poder modificar los archivos cuando encuentra conflictos o cuando usamos el comando `$ git commit` sin la bandera y el argumento: `-m "descripcion del commit"`. En caso de querer cambiar el editor por defecto disponemos del siguiente comando:

```
$ git config --global core.editor "el comando para abrir tu editor"
```

En mi caso voy a colocar el editor neovim, entonces el comando sería:

```
$ git config --global core.editor "nvim"
```

1.5. Comprobar la configuración de GIT

Para poder ver la configuración que tenemos en git podemos usar el siguiente comando:

```
$ git config --list
```

Que nos producira una salida de este tipo:

```
user.email=202301300@est.umss.edu
user.name=Jaime Sebastian Chavarria Fuertes
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

Podemos jugar con las banderas de `list`, por ejemplo:

- Para poder mostrar la unicamente la configuración global:

```
$ git config --global --list
```

- Para poder mostrar la configuración del repositorio local:

```
$ git config --local --list
```

Y muchas mas que se pueden revisar en este enlace de la documentación de GIT

1.6. Cómo inicializar un nuevo proyecto en GIT?

Para crear un nuevo repositorio local podemos utilizar el comando:

```
$ git init <direccion de la carpeta>
```

Este comando nos creara una carpeta con el nombre asignado en la dirección.

Pero si en caso de tener ya un proyecto podemos ejecutar el mismo comando dentro de la carpeta raíz de nuestro proyecto pero sin pasarle de argumento nada.

```
$ git init
```

Y asi ya habremos comenzado a utilizar GIT al fin.