



# git

## Nociones de GIT - Academia Java

### Descripción de Git

Git es una herramienta que permite a los desarrolladores tener un mayor control sobre los proyectos, específicamente es un **controlador de versiones**. El cual funciona como un sistema que **registra los cambios realizados en un archivo** o conjunto de archivos a lo largo del tiempo con el fin de llevar un registro de cambios, todo esto con el objetivo de poder regresar a una versión anterior del proyecto.

Es importante mencionar que Git trabaja de manera local, mientras que GitHub lo hace de manera remota a través de una plataforma en línea y permite almacenar proyectos basados en Git. Podemos conectar Git a diferentes tipos de controladores de versiones remotos como GitLab o GitHub.

Una de las principales Ventajas de Git son las ramas (branch), ya que permiten trabajar de forma simultanea donde cada rama representa una linea de desarrollo independiente. Todo esto de manera aislada lo cual otorga una gran ventaja ya que no podemos hacer cambios sin afectar la rama principal.

### Áreas de trabajo en Git

Git cuenta con tres áreas de trabajo donde se encuentran nuestros archivos.

- Working Directory: Área donde agregamos, editamos o eliminamos archivos.
- Staging Area: Área donde seleccionamos y preparamos los cambios.
- .git directory(Repository): Es donde se almacenan los metadatos y la base de datos de objetos.

## Guía Básica de Git.

### Configuración inicial.

```
# Configurar Nombre de Usuario
git config -- global user.name "nombredeusuario"

# Configurar correo electronico
git config --global user.email "email@deusuario.com"
```

### Comandos Básicos de Git

```
git init # Indica la creacion de un nuevo repositorio
```

En el caso de que se tenga un repositorio ya existente.

```
git clone <URL_del_Repositorio> # Clona un repositorio
```

Muestra el estado actual de nuestros archivos.

```
git status
```

Realizar cambios y confirmarlos.

```
# Añade los archivos al Stage Area lo esta preparando para ser c
git add <nombre_archivo>

# Confirmar cambios -> Pasar al area de repositorio
```

```
git commit -m "Cambios Agregados" # Confirma los cambios
```

Ver Historial de commits realizados

```
git log
```

Creando, cambiando y fusionando ramas.

```
# Crea una nueva rama  
git branch <nombre_rama>
```

```
# Crea una nueva rama  
git checkout <nombre_rama>
```

```
#Fusionar Ramas
```

```
git merge <nombre_rama> # Fusiona la rama especificada con la rama actual
```

## Diferencias

Mostrar cambios realizados en archivos

```
git diff # cambios realizados.
```

```
# ver diferencias entre el área de preparación y el último commit  
git diff --staged
```

```
# ver cambios entre dos commits  
git diff <commit_1> <commit_2>
```

## Deshacer Cambios

Deshacer cambios locales no confirmados.

```
git restore <nombre_archivo>
```

Deshacer cambios locales que fueron confirmados.

```
git reset --hard HEAD
```

## Trabajando con Repositorios Remotos (GitHub)

Conectar Repositorio Local con Repositorio Remoto.

```
git remote add origin <URL_del_Repositorio>
```

Subir cambios de repositorio local al remoto.

```
git push -u origin <nombre_rama>
```

Descargar cambios del repositorio remoto:

```
git pull origin <nombre_rama>
```

---

David Andres - Notion