

# Taller 1: Conociendo Git y Github

¿Qué es Git?



Git es un software de control de versiones diseñado por **Linus Torvalds**, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

¿Qué es Git?



Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

## Comandos principales.

A continuación se mencionarán los comandos básicos de esta herramienta de versionado denominada **git**.



### Comandos Git

#### Git init

```
git init
```

Permite crear la carpeta oculta `.git` y todo el sistema de versionado que lo acompaña.

#### Git Status

```
git status
```

Es el comando que más utilizado en este sistema de versionado, permite conocer el estado actual del proyecto.

---

### Comandos Git

#### Git add

```
git add archivo.extensión
```

El comando **add** permite agregar el o los archivos modificados al estado **staging**.

### Git rm

```
git rm archivo.extensión
```

El comando **rm** permite eliminar el archivo del repositorio de git.

---

## Comandos Git

### Git commit

```
git commit -m "mensaje"
```

Este comando permite pasar del estado de **staging** al repositorio (ya conoceremos los repositorios).

---

## Comandos principales.

### Git diff

```
git diff hash1 hash2
```

Este comando permite comparar una versión del repositorio con otro.

### Git log

```
git log archivo.extensión  
git log --stat
```

Permite conocer información mucho más enriquecida que **git status**.

---

## Comandos principales.

### Git reset

```
git reset hash
```

Permite regresar (como en una máquina del tiempo) al estado anterior del avance del proyecto.

Se recomienda utilizar si es necesario recuperar información que es necesaria en un punto donde no hay un error o no se ha dañado el proyecto.

---

## Comandos principales.

### Git reset

```
git reset hard
```

Consultar este comando.

```
git reset hard --soft
```

Consultar este comando.

---

## Comandos principales.

### Git checkout

```
git checkout hash
```

Según Molina Ríos et al. [\[1\]](#) este comando permite cambiar de un estado a otro (Siguiendo la máquina del tiempo), para conocer que se ha avanzado en el proyecto de acuerdo a cada **commit**.

```
git checkout main archivo.extensión
```

Este comando permite regresar al último commit realizado en el repositorio del proyecto.

## Estados de Git

---

Estados de Git.

Git Staging



git

Git Staging



git

---

Estados de Git.

Git Staging



git

## Git Staging



## Práctica.

---

### Práctica.

Sigue las siguientes indicaciones:

1. Crear un Directorio con el nombre Currículo Vitae.
2. Utilizar el comando `git init`.

```
git init
```

3. Crear un archivo con el nombre `**index.md**`

```
touch index.md
```

4. Abrir el archivo con VScode.
- 

### Práctica.

5. Escribir los datos principales de tu hoja de vida (Nombres y Apellidos, Correo, Teléfono, etc).
6. Agregar una foto de Perfil.

8. Agregar tu experiencia laboral (Si tiene certificados laborales por favor agregalos).

9. Agregar tus habilidades (Utiliza una lista no ordenada).

10. Agrega referencias personales.

## Referencias

## References

- [1] Jimmy Rolando Molina Ríos et al. “Evaluación de Los Frameworks En El Desarrollo de Aplicaciones Web Con Python”. In: *Revista Latinoamericana de Ingenieria de Software* 4.4 (Sept. 2016), p. 201. ISSN: 2314-2642. DOI: [10.18294/relais.2016.201-207](https://doi.org/10.18294/relais.2016.201-207).