

# Git & GitHub - Cheat Sheet

## Conceptos Básicos

**Git:** Sistema de control de versiones (en tu computadora)

**GitHub:** Plataforma en la nube para guardar y compartir código

**Repositorio:** Carpeta de proyecto con historial de cambios

**Commit:** 'Guardar' una versión del código con mensaje descriptivo

## Configuración Inicial (Solo una vez)

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu@email.com"  
git config --list
```

## Comandos Esenciales

### Iniciar un proyecto

```
git init  
git clone https://github.com/usuario/repositorio.git
```

### Flujo básico de trabajo

1. Ver estado de archivos:

```
git status
```

2. Agregar archivos al staging:

```
git add archivo.py  
git add .
```

3. Hacer commit:

```
git commit -m "Mensaje descriptivo del cambio"
```

4. Subir cambios a GitHub:

```
git push
```

5. Traer cambios de GitHub:

```
git pull
```

## Comandos de Información

```
git log  
git log --oneline  
git diff
```

## Branches (Ramas)

```
git branch  
git branch nombre-rama  
git checkout nombre-rama  
git checkout -b nombre-rama  
git merge nombre-rama
```

## Deshacer Cambios

```
git checkout -- archivo.py  
git reset archivo.py  
git show
```

## Trabajar con GitHub

### Primera vez (conectar proyecto local)

```
git remote add origin https://github.com/usuario/repositorio.git  
git branch -M main  
git push -u origin main
```

### Flujo diario

```
git pull  
git add .  
git commit -m "Descripción del cambio"  
git push
```

## Errores Comunes

Problema	Solución
fatal: not a git repository	Ejecuta git init primero
No puedo hacer push	Primero haz git pull
Conflictos al hacer pull	Resuelve conflictos manualmente, luego commit
Olvidé hacer commit antes de pull	Guarda cambios con git stash, luego pull



## Buenos Mensajes de Commit

### MAL:

```
git commit -m "cambios"  
git commit -m "fix"
```

### BIEN:

```
git commit -m "Aregar función de login"  
git commit -m "Corregir error en cálculo de promedio"  
git commit -m "Actualizar README con instrucciones"
```

## Comandos Más Usados (Top 10)

1. git status - Ver estado actual
2. git add . - Preparar todos los cambios
3. git commit -m 'mensaje' - Guardar versión
4. git push - Subir a GitHub
5. git pull - Traer cambios de GitHub
6. git clone URL - Copiar repositorio
7. git log - Ver historial
8. git branch - Ver/crear ramas
9. git checkout - Cambiar de rama
10. git init - Iniciar repositorio

## Archivo .gitignore

Crear archivo .gitignore para NO subir ciertos archivos:

```
# Archivos Python  
__pycache__/  
*.pyc  
*.pyo  
venv/  
env/  
  
# Archivos del sistema  
.DS_Store  
Thumbs.db  
  
# Archivos de configuración  
config.py  
.env
```

## Ayuda Rápida

```
git help  
git help commit  
git commit --help
```

Tip Final: Practica estos comandos frecuentemente. Git se aprende usando.