

## GUIA DE COMANDOS GIT - REFERENCIA RÁPIDA

### 1. SUBIR UN REPOSITORIO POR PRIMERA VEZ

#### Opción A: Repositorio nuevo en GitHub (ya creado en web)

```
bash
# 1. Inicializar Git en tu carpeta local
git init

# 2. Añadir todos los archivos
git add .

# 3. Hacer el primer commit
git commit -m "Primer commit"

# 4. Añadir el repositorio remoto
git remote add origin https://github.com/TU_USUARIO/TU_REPO.git

# 5. Cambiar a la rama main (si es necesario)
git branch -M main

# 6. Subir al repositorio remoto
git push -u origin main
```

#### Opción B: Crear repositorio desde cero

```
bash
# 1. Crear repositorio en GitHub primero (desde la web)

# 2. Clonar el repositorio vacío
git clone https://github.com/TU_USUARIO/TU_REPO.git

# 3. Entrar en la carpeta
cd TU_REPO

# 4. Añadir tus archivos y hacer commit
git add .
git commit -m "Primer commit"

# 5. Subir los cambios
git push origin main
```

### 2. ACTUALIZAR ARCHIVOS EN EL REPOSITORIO (Subir cambios)

```
bash
# 1. Ver qué archivos han cambiado
git status

# 2. Añadir los archivos modificados
git add .          # Añade TODOS los archivos
# O específicos:
git add archivo1.py archivo2.txt # Añade archivos específicos

# 3. Hacer commit con mensaje descriptivo
git commit -m "Descripción de los cambios realizados"

# 4. Subir los cambios al repositorio remoto
git push origin main

# NOTA: Si hay conflictos, primero hacer pull (ver sección 4)
```

#### ⚡ Atajo rápido (todo en uno):

```
bash
git add . && git commit -m "Actualización" && git push origin main
```

### 3. DESCARGAR UN REPOSITORIO POR PRIMERA VEZ

```
bash
# Clonar el repositorio completo en tu PC
git clone https://github.com/USUARIO/REPOSITORIO.git

# Esto crea una carpeta con el nombre del repositorio
# y descarga todos los archivos

# Ejemplo real:
git clone https://github.com/idalonso123/PROGRAMA_PEDIDOS.git

# Luego entrar en la carpeta:
cd PROGRAMA_PEDIDOS
```

### 4. ACTUALIZAR ARCHIVOS LOCALES (Descargar cambios)

```
bash
# Opción 1: Traer y fusionar cambios automáticamente
git pull origin main

# Opción 2: Traer cambios pero revisar antes de fusionar
git fetch origin      # Descarga cambios sin aplicarlos
git merge origin/main # Aplica los cambios descargados
```

#### ⚠ Si hay CONFLICTOS al hacer pull:

```
bash
# 1. Hacer pull
git pull origin main

# 2. Si hay conflictos, Git te avisará
# Resolver conflictos manualmente en los archivos

# 3. Después de resolver, añadir los archivos
git add .

# 4. Completar el merge
git commit -m "Resueltos conflictos"

# 5. Subir los cambios
git push origin main
```

#### 💡 Estrategias para resolver conflictos rápidamente:

```
bash
```

```

# Mantener SIEMPRE tu versión local (sobrescribe remota)
git checkout --ours ARCHIVO
git add ARCHIVO

# Mantener SIEMPRE la versión remota (descarta local)
git checkout --theirs ARCHIVO
git add ARCHIVO

# Para TODOS los archivos en conflicto:
git checkout --ours . # Mantener todo local
# o
git checkout --theirs . # Mantener todo remoto

```

## ⌚ COMANDOS ÚTILES ADICIONALES

Ver estado actual:

```

bash
git status          # Ver archivos modificados
git log             # Ver historial de commits
git log --oneline   # Ver historial resumido

```

Ver diferencias:

```

bash
git diff            # Ver cambios no guardados
git diff ARCHIVO    # Ver cambios en archivo específico

```

Deshacer cambios:

```

bash
git restore ARCHIVO  # Descartar cambios locales
git restore .         # Descartar TODOS los cambios
git reset --hard      # Volver al último commit (/CUIDADO!)

```

Trabajar con ramas:

```

bash
git branch           # Ver ramas locales
git branch NOMBRE    # Crear nueva rama
git checkout NOMBRE   # Cambiar a otra rama
git merge NOMBRE     # Fusionar rama en la actual

```

Información del repositorio:

```

bash
git remote -v        # Ver repositorios remotos configurados
git branch -a         # Ver todas las ramas (locales y remotas)

```

## ⌚ WORKFLOW TÍPICO DIARIO

Al empezar a trabajar:

```

bash
git pull origin main  # Descargar últimos cambios

```

Mientras trabaja:

```

bash
git status           # Ver qué has modificado

```

Al terminar tu trabajo:

```

bash
git add .
git commit -m "Descripción clara de los cambios"
git push origin main

```

## ⌚ SITUACIONES COMUNES Y SOLUCIONES

"No puedo hacer push porque el remoto tiene cambios"

```

bash
git pull origin main  # Traer cambios primero
# Resolver conflictos si los hay
git push origin main  # Luego subir

```

"Quiero sobrescribir el remoto con mi versión local"

```

bash
git push --force origin main  # CUIDADO: Elimina cambios remotos

```

"Cloné el repo pero quiero ver otra rama"

```

bash
git checkout NOMBRE_RAMA  # Cambiar a otra rama existente

```

"Eliminé archivos por error"

```

bash
git restore .             # Recuperar archivos del último commit

```

## ⌚ RESUMEN ULTRA-RÁPIDO

Acción	Comando
Primera vez subir	(git init) → (git add.) → (git commit -m "msg") → (git remote add origin URL) → (git push -u origin main)
Actualizar repo	(git add.) → (git commit -m "msg") → (git push origin main)
Primera vez descargar	(git clone URL)
Actualizar local	(git pull origin main)

## ⌚ ENLACES ÚTILES

- Documentación oficial Git: <https://git-scm.com/doc>
- GitHub Docs: <https://docs.github.com/es>
- Tutorial interactivo: <https://learngitbranching.js.org/>

