Manual Detallado de Git en Bash (con Solución a Autenticación)

Pensado para principiantes: pasos claros, ejemplos y comandos listos para copiar/pegar.

1) Flujo básico para subir cambios

1. **1. Ubícate en la carpeta del proyecto:** Abre el Explorador, clic derecho » *Git Bash Here*.
2. **2. Verifica el estado: git status** (archivos nuevos = *untracked*, modificados = *modified*).
3. **3. Agrega los cambios: git add .** (todo) o **git add archivo.java** (uno específico).
4. **4. Crea el commit: git commit -m "Describe el cambio"**.
5. **5. Sube al remoto: git push -u origin main** (si tu rama principal es *master*, usa **origin master**).
6. **6. Confirmar en GitHub:** entra a tu repo y verifica los archivos/commits.

Comandos esenciales (día a día)

Comando	¿Para qué sirve?
git status	Ver qué cambió y qué está en staging.
git add .	Agregar TODOS los cambios al staging.
git add <archivo>	Agregar un archivo puntual al staging.
git commit -m "msg"	Guardar cambios en el historial local.
git push	Enviar tus commits al remoto.
git pull	Traer y fusionar cambios del remoto.
git log --oneline	Ver historial simplificado.
git diff	Ver diferencias antes del add/commit.

2) Ramas: crear, cambiar y fusionar

- **Crear una rama: git branch feat/menu** (o) **git switch -c feat/menu**
- **Cambiar de rama: git checkout feat/menu** (o) **git switch feat/menu**
- **Publicar una rama nueva: git push -u origin feat/menu**
- **Volver a main y actualizarla: git checkout main** y **git pull**
- **Fusionar (merge):** estando en **main**, ejecuta **git merge feat/menu**
- **Resolver conflictos:** abre los archivos con marcas <<<<<<< ======= >>>>>>>, edita, guarda, luego **git add** y **git commit**.

3) .gitignore en pocas palabras

Evita subir archivos innecesarios/sensibles. Crea un archivo **.gitignore** con líneas como:

```
target/
*.log
.env
.vscode/
__pycache__/
```

4) Problemas de autenticación (HTTPS) — Soluciones paso a paso

Si ves errores como *"Password authentication is not supported for Git operations"* o *"Authentication failed"*, usa uno de estos caminos.

A) Inicio de sesión por navegador (recomendado)

- Configura el gestor de credenciales de Git (una única vez):
- **git config --global credential.helper manager**
- Si tenías credenciales viejas guardadas, bórralas:
- **git credential-manager erase https://github.com**
- Realiza el push nuevamente para que se abra el navegador y autorices:
- **git push -u origin main**
- Completa el login/consentimiento en el navegador. Vuelve a la terminal: el push debería terminar correctamente.

B) Usar un Personal Access Token (PAT)

- En GitHub: Settings → Developer settings → Personal access tokens → Tokens (classic) → Generate new token.
- Marca al menos el alcance **repo**. Copia el token (se muestra una sola vez).
- En Git Bash, ejecuta el push: **git push -u origin main**
- Cuando pida credenciales: Usuario = tu usuario de GitHub. Contraseña = pega el **token** (no tu contraseña normal).
- Si quieres borrar un token guardado (por seguridad): **git credential-manager erase https://github.com**.

C) Ver o corregir el remoto (origin)

A veces el error es la URL mal configurada:

```
Ver remotos: git remote -v
Cambiar URL: git remote set-url origin https://github.com/USUARIO/REPO.git
Agregar origin si no existe: git remote add origin https://github.com/USUARIO/REPO.git
```

D) Tu repo usa master en vez de main

Si el remoto se llama *master*:

```
git branch -M master
git push -u origin master
```

E) Alternativa: usar SSH (opcional)

- Genera una llave: **ssh-keygen -t ed25519 -C "tu-email"** (presiona Enter a todo).
- Copia la clave pública (~/.ssh/id_ed25519.pub) y agrégala en GitHub → Settings → SSH and GPG keys.
- Cambia el remoto a SSH: **git remote set-url origin git@github.com:USUARIO/REPO.git**
- Prueba conexión: **ssh -T git@github.com**. Luego **git push** normalmente (sin usuario/contraseña).

5) Diagnóstico rápido cuando algo no sube

- **¿Estás en la carpeta correcta?** Ejecuta **git rev-parse --show-toplevel**.
- **¿Hay cambios para subir? git status**. Si no hay nada en staging ni commits, no habrá push.
- **¿Tu rama está vinculada al remoto? git branch -vv**. Si no, usa **git push -u origin tu-rama**.
- **¿El remoto existe? git remote -v**. Corrige con **git remote set-url origin ...**.
- **¿Conflictos al hacer pull?** Resuélvelos, **git add** y **git commit**, luego **git push**.
- **¿Error de autenticación?** Usa la sección 4 (A o B).

6) Alias útiles (ahorran tiempo)

```
git config --global alias.lg "log --oneline --graph --decorate --all"
git config --global alias.st "status -sb"
git config --global alias.co checkout
git config --global alias.ci commit
```