

Sesión 2: HTTPS vs SSH

Elige el método de autenticación adecuado para tus repos: HTTPS (tokens) o SSH (claves), entendiendo seguridad, fricción y contexto de uso.

Objetivos

- Diferenciar HTTPS y SSH para acceder a repos.
- Configurar el método según entorno y políticas.
- Cambiar entre métodos sin afectar el historial.

Requisitos

- Git instalado (Sesión 1).
- Cuenta en GitHub/GitLab/Bitbucket.

HTTPS vs SSH (rápido)

- HTTPS
 - Pros: sencillo, funciona tras proxies; usa PAT/tokens.
 - Contras: más prompts si no usas gestor de credenciales.
- SSH
 - Pros: sin contraseñas, fluido, seguro con claves.
 - Contras: requiere generar/registrar claves; a veces bloqueado por firewalls.

URLs por proveedor

- GitHub: HTTPS `https://github.com/ORG/REPO.git` | SSH `git@github.com:ORG/REPO.git`
- GitLab: HTTPS `https://gitlab.com/ORG/REPO.git` | SSH `git@gitlab.com:ORG/REPO.git`
- Bitbucket: HTTPS `https://bitbucket.org/ORG/REPO.git` | SSH `git@bitbucket.org:ORG/REPO`

Recomendaciones de uso

- Empresa con proxy/SAML: HTTPS + Git Credential Manager (GCM).
- Desarrollo diario multiproyecto: SSH (ver Sesión 4).
- CI/CD: SSH o PAT con permisos mínimos.

Gestores de credenciales (cuando uses HTTPS)

- Linux (ejemplos)

```
# Recomendado: GCM (ver Sesión 3)
# Alternativa: libsecret/gnome-keyring
sudo apt install -y libsecret-1-0 libsecret-1-dev 2>/dev/null || true
# Configura el helper (usa uno)
git config --global credential.helper manager-core
# o
# git config --global credential.helper /usr/share/doc/git/contrib/credential/libsecret
```

- Windows

```
# GCM viene con Git for Windows
git config --global credential.helper manager
```

- macOS

```
# GCM o llavero de macOS
git config --global credential.helper manager-core
# o
# git config --global credential.helper osxkeychain
```

Cambiar el remoto

```
cd mi-repo
git remote -v
# Cambiar de HTTPS a SSH
git remote set-url origin git@github.com:ORG/REPO.git
# O de SSH a HTTPS
git remote set-url origin https://github.com/ORG/REPO.git
git remote -v
```

Verificación

- HTTPS: `git fetch` debería funcionar sin pedir credenciales repetidas si GCM está bien.
- SSH: prueba con `ssh -T git@github.com`, `ssh -T git@gitlab.com`, `ssh -T git@bitbucket.org`.

Políticas por proveedor (resumen)

- GitHub: PAT para HTTPS (no contraseñas), reglas de organización.
- GitLab: tokens personales y SSH por usuario; caducidad configurable.
- Bitbucket: tokens de app y SSH por usuario/equipo.

Problemas comunes

- Proxy corporativo: configura `https.proxy/http.proxy` en Git y el sistema.
- SSH bloqueado: usa HTTPS o SSH por 443 si el proveedor lo ofrece.
- Varias cuentas: usa `~/.ssh/config` (Sesión 4) o múltiples perfiles con GCM (Sesión 3).