# Sesión 4: Claves SSH

Autentícate con repositorios sin contraseñas usando pares de claves SSH (más seguro y fluido que contraseñas).

## Qué vas a lograr (en 5 pasos)

1. Generar una clave SSH moderna (ed25519) protegida con passphrase.
2. Cargar la clave en el agente (para no reingresar la passphrase todo el tiempo).
3. Registrar la clave pública en GitHub, GitLab y/o Bitbucket.
4. Probar la conexión.
5. (Opcional) Configurar múltiples identidades con `~/.ssh/config`.

## Requisitos

- OpenSSH instalado (Linux y macOS lo traen; Windows 10+ lo incluye como característica).
- Acceso a tu cuenta del proveedor (GitHub/GitLab/Bitbucket).

Sugerencia: Verifica la versión de OpenSSH

```
ssh -V
```

---

## Paso 1: Generar clave SSH

- Linux (bash)

```
ssh-keygen -t ed25519 -C "tu.email@example.com"
# Ruta sugerida: /home/USUARIO/.ssh/id_ed25519
```

- Windows (PowerShell)

```
ssh-keygen -t ed25519 -C "tu.email@example.com"
# Ruta sugerida: C:\Users\USUARIO\.ssh\id_ed25519
```

- macOS (bash/zsh)

```
ssh-keygen -t ed25519 -C "tu.email@example.com"
# Ruta sugerida: /Users/USUARIO/.ssh/id_ed25519
```

Recomendado: usa una passphrase. Si necesitas compatibilidad amplia, RSA-4096 también es válido, pero ed25519 es más moderno y rápido.

---

### Paso 2: Cargar la clave en el agente SSH

- Linux (bash)

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

- Windows (PowerShell)

```
Start-Service ssh-agent
Get-Service ssh-agent | Set-Service -StartupType Automatic
ssh-add $env:USERPROFILE\.ssh\id_ed25519
```

- macOS (bash/zsh)

```
eval "$(ssh-agent -s)"
ssh-add --apple-use-keychain ~/.ssh/id_ed25519
```

Notas

- Si `ssh-add` pide passphrase, es normal (es la de tu clave privada).
- En macOS, `--apple-use-keychain` guarda la passphrase en el llavero.

---

### Paso 3: Registrar tu clave pública en el proveedor

Primero, copia tu clave pública:

- Linux

```
cat ~/.ssh/id_ed25519.pub
```

- Windows (PowerShell)

```
type $env:USERPROFILE\.ssh\id_ed25519.pub
```

- macOS

```
pbcopy < ~/.ssh/id_ed25519.pub  # copia al portapapeles
```

Luego, pégala en tu proveedor:

- GitHub: Settings → SSH and GPG keys → New SSH key.
- GitLab: Preferences → SSH Keys → Add SSH Key.
- Bitbucket: Personal settings → SSH keys → Add key.

---

### Paso 4: Probar la conexión

```
ssh -T git@github.com
ssh -T git@gitlab.com
ssh -T git@bitbucket.org
```

Deberías ver un mensaje de bienvenida (o confirmación de autenticidad del host la primera vez).

---

### Paso 5 (opcional): Múltiples cuentas con `~/.ssh/config`

- Linux y macOS (bash)

```
cat >> ~/.ssh/config <<'EOF'
Host github-personal
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_ed25519
  IdentitiesOnly yes

Host gitlab-work
  HostName gitlab.com
  User git
  IdentityFile ~/.ssh/id_ed25519_work
  IdentitiesOnly yes
EOF
```

- Windows: edita `C:\Users\USUARIO\.ssh\config` con contenido equivalente.

Usa el alias en la URL SSH, por ejemplo: `git@github-personal:ORG/REPO.git`.

---

### Solución de problemas

- Permisos de archivo (Linux/macOS):

```
chmod 600 ~/.ssh/id_ed25519
chmod 700 ~/.ssh
```

- El agente no carga la clave: vuelve a ejecutar `ssh-agent` y `ssh-add`.
- Firewalls que bloquean puerto 22: revisa si el proveedor ofrece SSH por 443.
- Host key changed warning: podría ser un ataque o un cambio real del proveedor; valida en el estado del proveedor antes de continuar.