Sesión 5: Firma GPG (GNU Privacy Guard)

Firma tus commits y tags para aportar autenticidad e integridad; habilita la verificación en el proveedor.

Qué vas a lograr

- Generar un par de claves GPG.
- Configurar Git para firmar automáticamente.
- Registrar la clave pública en GitHub/GitLab/Bitbucket.

Requisitos

- Git instalado; posibilidad de instalar GnuPG.

Instalar GnuPG

- Linux

```
sudo apt update && sudo apt install -y gnupg pinentry-gnome3 || sudo dnf install
```

- Windows

```
winget install --id GnuPG.GnuPG -e
```

- macOS

```
brew install gnupg pinentry-mac
```

Paso 1: Generar y listar claves

```
gpg --full-generate-key
# Tipo sugerido: RSA y RSA (4096) o ECC si disponible
gpg --list-secret-keys --keyid-format=long
```

Identifica tu KEYID (ej.: ABCDEF1234567890).

Paso 2: Configurar Git para firmar

```
git config --global user.signingkey ABCDEF1234567890
git config --global commit.gpgsign true
git config --global gpg.program gpg
```

Paso 3: Exportar y registrar clave pública

```
gpg --armor --export ABCDEF1234567890
```

Copia la salida y pégala en:

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

Uso diario

```
# Commit firmado automáticamente
git commit -m "feat: mi cambio"
# Tag firmado
git tag -s v1.0.0 -m "Release 1.0.0"
git push --follow-tags
```

Problemas comunes

- No aparece pinentry: instala/configura `pinentry` (en macOS `pinentry-mac`) y verifica `gpg-agent`.
- Email no coincide: la identidad de GPG debe coincidir con `user.email` o estar añadida en el proveedor.
- CI: considera usar una clave dedicada o SSH signing si tu plataforma lo soporta.