



Generar Clave SSH Para GitHub



Guía Completa Paso a Paso

Esta guía te ayudará a generar una nueva clave SSH y configurarla con GitHub para poder hacer push sin problemas.



Método 1: Generar Nueva Clave SSH (Recomendado)

Paso 1: Conectar a Tu Servidor

```
ssh usuario@tu-servidor.com
```

Paso 2: Generar la Clave SSH

```
# Generar clave ED25519 (más moderna y segura)
ssh-keygen -t ed25519 -C "tu-email@ejemplo.com" -f ~/.ssh/escalafin_github

# O si tu sistema no soporta ED25519, usa RSA:
ssh-keygen -t rsa -b 4096 -C "tu-email@ejemplo.com" -f ~/.ssh/escalafin_github
```

Notas:

- Presiona Enter cuando pregunte por passphrase (para sin contraseña)
- O ingresa una contraseña si prefieres más seguridad

Salida esperada:

```
Generating public/private ed25519 key pair.
Your identification has been saved in /home/ubuntu/.ssh/escalafin_github
Your public key has been saved in /home/ubuntu/.ssh/escalafin_github.pub
The key fingerprint is:
SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx tu-email@ejemplo.com
```

Paso 3: Ver la Clave Pública

```
cat ~/.ssh/escalafin_github.pub
```

Copia TODO el contenido que se muestra. Debería verse así:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx tu-
email@ejemplo.com
```

O para RSA:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX tu-
email@ejemplo.com
```

Paso 4: Agregar la Clave a GitHub

1. **Ir a GitHub SSH Keys:**
<https://github.com/settings/keys>
2. **Click en “New SSH key”** (botón verde)
3. **Rellenar el formulario:**
 - **Title:** EscalaFin Server (o el nombre que prefieras)
 - **Key type:** Authentication Key
 - **Key:** Pega la clave pública completa que copiaste
4. **Click en “Add SSH key”**
5. **Confirmar con tu contraseña de GitHub** si te lo pide

Paso 5: Configurar SSH en el Servidor

Crear/editar el archivo de configuración SSH:

```
# Crear archivo de configuración
cat > ~/.ssh/config << 'EOF'
Host github.com
    HostName github.com
    User git
    IdentityFile ~/.ssh/escalafin_github
    IdentitiesOnly yes
EOF

# Dar permisos correctos
chmod 600 ~/.ssh/config
chmod 600 ~/.ssh/escalafin_github
chmod 644 ~/.ssh/escalafin_github.pub
```

Paso 6: Probar la Conexión

```
ssh -T git@github.com
```

Respuesta esperada:

```
Hi ghositing! You've successfully authenticated, but GitHub does not provide shell access.
```

Si es la primera vez, verás:

```
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
Are you sure you want to continue connecting (yes/no)?
```

Responde: yes

Paso 7: Hacer Push

```
cd /home/ubuntu/escalafin_mvp
git push origin main
```

Método 2: Usar Clave SSH Existente

Si ya tienes una clave SSH en tu servidor:

Paso 1: Ver Claves Existentes

```
ls -la ~/.ssh/
```

Busca archivos como:

- `id_ed25519` y `id_ed25519.pub`
- `id_rsa` y `id_rsa.pub`
- Cualquier archivo `.pub`

Paso 2: Ver Clave Pública

```
# Para ED25519
cat ~/.ssh/id_ed25519.pub

# Para RSA
cat ~/.ssh/id_rsa.pub

# O la que tengas
cat ~/.ssh/NOMBRE_DE_TU_CLAVE.pub
```

Paso 3: Copiar y Agregar a GitHub

Sigue el **Paso 4** del Método 1 para agregar la clave a GitHub.

Paso 4: Configurar Git

```
# Si usas una clave con nombre personalizado
cat > ~/.ssh/config << 'EOF'
Host github.com
    HostName github.com
    User git
    IdentityFile ~/.ssh/NOMBRE_DE_TU_CLAVE
    IdentitiesOnly yes
EOF

chmod 600 ~/.ssh/config
```

Paso 5: Probar

```
ssh -T git@github.com
```

Método 3: Script Automatizado

Copia y pega este script completo:

```
#!/bin/bash

echo "🔑 Configuración SSH para GitHub - EscalaFin"
echo "=====
echo

# Variables
EMAIL="escalafin@qhosting.com"
KEY_NAME="escalafin_github"
KEY_PATH="$HOME/.ssh/$KEY_NAME"

# 1. Verificar si ya existe una clave
if [ -f "$KEY_PATH" ]; then
    echo "⚠️ Ya existe una clave en: $KEY_PATH"
    read -p "¿Deseas generar una nueva? (s/n): " -n 1 -r
    echo
    if [[ ! $REPLY =~ ^[Ss]$ ]]; then
        echo "❌ Operación cancelada"
        exit 0
    fi
    mv "$KEY_PATH" "${KEY_PATH}.backup.${date +%Y%m%d_%H%M%S}"
    echo "✅ Clave anterior respaldada"
fi

# 2. Generar nueva clave
echo "🔑 Generando nueva clave SSH..."
ssh-keygen -t ed25519 -C "$EMAIL" -f "$KEY_PATH" -N "" 2>/dev/null || \
ssh-keygen -t rsa -b 4096 -C "$EMAIL" -f "$KEY_PATH" -N ""

if [ $? -eq 0 ]; then
    echo "✅ Clave generada exitosamente!"
else
    echo "❌ Error al generar la clave"
    exit 1
fi

# 3. Configurar SSH
echo
echo "⚙️ Configurando SSH..."
cat > ~/.ssh/config << EOF
Host github.com
    HostName github.com
    User git
    IdentityFile $KEY_PATH
    IdentitiesOnly yes
EOF

chmod 600 ~/.ssh/config
chmod 600 "$KEY_PATH"
chmod 644 "${KEY_PATH}.pub"

echo "✅ Configuración SSH completada"

# 4. Mostrar clave pública
echo
echo "
echo "
echo "
echo "📋 COPIA ESTA CLAVE Y AGRÉGALA A GITHUB"
echo "
echo "
echo
cat "${KEY_PATH}.pub"
```

```

echo
echo "
echo "
echo " 🔗 Agrégala aquí: https://github.com/settings/keys "
echo "
echo "
echo

# 5. Esperar confirmación
read -p "Presiona Enter después de agregar la clave a GitHub..."

# 6. Probar conexión
echo
echo " 🛠️ Probando conexión con GitHub..."
ssh -T git@github.com 2>&1 | grep -q "successfully authenticated" && \
    echo "✅ ;Conexión exitosa! Ya puedes hacer push" || \
    echo "⚠️ Conexión no exitosa. Verifica que hayas agregado la clave correctamente"

echo
echo " 🎯 Para hacer push ahora:"
echo "     cd /home/ubuntu/escalafin_mvp"
echo "     git push origin main"
echo

```

Para usar el script:

```

# Crear el script
nano ~/setup_github_ssh.sh

# Pegar el contenido del script
# Guardar: Ctrl+O, Enter, Ctrl+X

# Dar permisos
chmod +x ~/setup_github_ssh.sh

# Ejecutar
~/setup_github_ssh.sh

```

Troubleshooting

Error: “Permission denied (publickey)”

Causa: La clave no está agregada a GitHub o no está configurada correctamente.

Solución:

1. Verificar que la clave esté en GitHub:


```

```bash
Ver tu clave
cat ~/.ssh/escalafin_github.pub

```

```
Compararla con las claves en:
https://github.com/settings/keys
...
```

1. Verificar configuración SSH:

```
bash
cat ~/.ssh/config
```

Debería incluir:

```
Host github.com
 IdentityFile ~/.ssh/escalafin_github
```

1. Probar con verbose:

```
bash
ssh -vT git@github.com
```

Esto mostrará qué clave está intentando usar.

## Error: “Host key verification failed”

**Causa:** Primera conexión a GitHub desde este servidor.

**Solución:**

```
Agregar GitHub a known_hosts
ssh-keyscan github.com >> ~/.ssh/known_hosts

O conectar manualmente una vez
ssh -T git@github.com

Responde "yes" cuando pregunte
```

## Error: “Could not open a connection to your authentication agent”

**Causa:** El agente SSH no está corriendo.

**Solución:**

```
Iniciar el agente SSH
eval "$(ssh-agent -s)"

Agregar tu clave
ssh-add ~/.ssh/escalafin_github
```

## Error: “Bad owner or permissions”

**Causa:** Permisos incorrectos en archivos SSH.

**Solución:**

```
Arreglar permisos del directorio
chmod 700 ~/.ssh

Arreglar permisos de claves privadas
chmod 600 ~/.ssh/escalafin_github
chmod 600 ~/.ssh/config

Arreglar permisos de claves públicas
chmod 644 ~/.ssh/escalafin_github.pub
chmod 644 ~/.ssh/known_hosts
```

---

## Verificación Completa

---

Script para verificar toda la configuración:



```
#!/bin/bash

echo "🔍 Verificación de Configuración SSH para GitHub"
echo "===== "
echo

1. Verificar claves SSH
echo "1 Claves SSH disponibles:"
ls -lh ~/.ssh/*.pub 2>/dev/null || echo " ❌ No se encontraron claves públicas"
echo

2. Verificar configuración SSH
echo "2 Configuración SSH:"
if [-f ~/.ssh/config]; then
 echo "✅ ~/.ssh/config existe"
 echo "Contenido para github.com:"
 grep -A 4 "^Host github.com" ~/.ssh/config | sed 's/^/ /'
else
 echo "⚠️ No existe ~/.ssh/config"
fi
echo

3. Verificar permisos
echo "3 Permisos:"
echo " ~/.ssh: $(stat -c '%a' ~/.ssh 2>/dev/null || echo 'N/A')"
echo " ~/.ssh/config: $(stat -c '%a' ~/.ssh/config 2>/dev/null || echo 'N/A')"
echo

4. Test de conexión
echo "4 Test de conexión:"
ssh -T git@github.com 2>&1 | head -1
echo

5. Configuración Git
echo "5 Configuración Git:"
echo "Remote URL: $(cd /home/ubuntu/escalafin_mvp && git remote get-url origin 2>/dev/null || echo 'N/A')"
echo "Branch: $(cd /home/ubuntu/escalafin_mvp && git branch --show-current 2>/dev/null || echo 'N/A')"
echo

6. Commits pendientes
echo "6 Commits pendientes de push:"
cd /home/ubuntu/escalafin_mvp 2>/dev/null && git log origin/main..HEAD --oneline 2>/dev/null | head -5 || echo " N/A"
echo
```

## Recursos Adicionales

### Enlaces Útiles

Recurso	URL
GitHub SSH Keys	<a href="https://github.com/settings/keys">https://github.com/settings/keys</a>
Guía Oficial GitHub SSH	<a href="https://docs.github.com/en/authentication/connecting-to-github-with-ssh">https://docs.github.com/en/authentication/connecting-to-github-with-ssh</a>
Troubleshooting SSH	<a href="https://docs.github.com/en/authentication/troubleshooting-ssh">https://docs.github.com/en/authentication/troubleshooting-ssh</a>

### Comandos de Referencia Rápida

```
Generar clave ED25519
ssh-keygen -t ed25519 -C "email@ejemplo.com"

Ver clave pública
cat ~/.ssh/id_ed25519.pub

Probar conexión
ssh -T git@github.com

Ver configuración SSH
cat ~/.ssh/config

Listar claves SSH
ls -la ~/.ssh/

Agregar clave al agente
ssh-add ~/.ssh/nombre_clave

Ver claves en el agente
ssh-add -l

Push con SSH
git push origin main
```

## Checklist de Configuración

Verifica que hayas completado todos estos pasos:

- ☐ Generar clave SSH en el servidor
- ☐ Copiar clave pública
- ☐ Agregar clave a GitHub (<https://github.com/settings/keys>)
- ☐ Configurar ~/.ssh/config
- ☐ Verificar permisos (600 para privada, 644 para pública)
- ☐ Probar conexión: `ssh -T git@github.com`
- ☐ Ver mensaje: “successfully authenticated”

- [ ] Hacer push: `git push origin main`
- [ ] Verificar en GitHub que los commits aparezcan

---

## Resumen - Comandos Esenciales

---

Generar y configurar todo de una vez:

```
1. Generar clave
ssh-keygen -t ed25519 -C "escalafin@qhosting.com" -f ~/.ssh/escalafin_github -N ""

2. Ver clave pública (COPIAR ESTO)
cat ~/.ssh/escalafin_github.pub

3. Agregar a GitHub: https://github.com/settings/keys

4. Configurar SSH
cat > ~/.ssh/config << 'EOF'
Host github.com
 HostName github.com
 User git
 IdentityFile ~/.ssh/escalafin_github
 IdentitiesOnly yes
EOF


chmod 600 ~/.ssh/config
chmod 600 ~/.ssh/escalafin_github

5. Probar
ssh -T git@github.com

6. Push
cd /home/ubuntu/escalafin_mvp
git push origin main
```

---

 **Creado:** 16 de Octubre de 2025

 **Objetivo:** Configurar SSH para push a GitHub

 **Tiempo estimado:** 5 minutos

---

 **¡Listo!**

Una vez completados estos pasos, podrás hacer push sin problemas:

```
cd /home/ubuntu/escalafin_mvp && git push origin main
```

**¡Éxito!** 🎉