



# Guía para Actualizar GitHub



## Resumen

Tienes **4 commits locales** pendientes de subir a GitHub:

```
86c1ed6 Multi-instance deployment & local testing
d641960 Production deployment tested and verified
f124ba2 Multi-instance Coolify deployment ready
309e2d5 47049d1c-8047-461e-8d06-585240fa0e88
```



**Repositorio:** <https://github.com/qhosting/escalafin-mvp>



## Opción 1: Script Automático (Recomendado)

El script más simple para subir los cambios:

```
cd /home/ubuntu/escalafin_mvp
./push-to-github.sh
```

**Cuando te pida credenciales:**

- **Username:** tu-usuario-de-github
- **Password:** tu-personal-access-token (NO tu password real)



## Opción 2: Comando Manual

Si prefieres hacerlo manualmente:

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



## Opción 3: Bundle de Git (Sin credenciales)

Si no tienes acceso a credenciales ahora, genera un bundle:

```
cd /home/ubuntu/escalafin_mvp
git bundle create escalafin-update-$(date +%Y%m%d).bundle origin/main..HEAD
```

Luego, desde una máquina con acceso a GitHub:

```
# Clonar el repositorio
git clone https://github.com/qhosting/escalafin-mvp.git
cd escalafin-mvp

# Aplicar el bundle
git pull /path/to/escalafin-update-YYYYMMDD.bundle main

# Push
git push origin main
```

---

## Crear Personal Access Token (PAT)

---

Si no tienes un PAT, créalo así:

### Paso 1: Ir a GitHub Settings

Visita: <https://github.com/settings/tokens>

### Paso 2: Generate New Token

1. Click en **“Generate new token”** → **“Generate new token (classic)”**
2. Dale un nombre descriptivo: `EscalaFin Deploy`
3. Establece expiración según tu preferencia (recomendado: 90 días)

### Paso 3: Seleccionar Scopes

Marca estas opciones:

- ☒ **repo** (Full control of private repositories)
- ☒ repo:status
- ☒ repo\_deployment
- ☒ public\_repo
- ☒ repo:invite

### Paso 4: Generar y Copiar

1. Click en **“Generate token”**
2. **¡IMPORTANTE!** Copia el token inmediatamente (no podrás verlo después)
3. Guárdalo en un lugar seguro

### Paso 5: Usar el Token

Cuando Git te pida password, usa el **token** (no tu password de GitHub):

```
Username: tu-usuario-de-github
Password: ghp_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

---

## Opción 4: Configurar SSH (Método Permanente)

---

Si quieres evitar usar tokens cada vez, configura SSH:

## Generar clave SSH

```
ssh-keygen -t ed25519 -C "tu-email@example.com"
```

## Agregar clave a ssh-agent

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

## Copiar clave pública

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

## Agregar a GitHub

1. Ve a: <https://github.com/settings/keys>
2. Click en **"New SSH key"**
3. Pega la clave pública
4. Dale un título descriptivo

## Cambiar remote a SSH

```
cd /home/ubuntu/escalafin_mvp  
git remote set-url origin git@github.com:qhosting/escalafin-mvp.git  
git push origin main
```



## Verificar Subida Exitosa

Después de hacer push, verifica:

```
cd /home/ubuntu/escalafin_mvp  
git log origin/main..HEAD
```

Si no muestra commits, ¡todo está sincronizado!

También puedes verificar en GitHub:

<https://github.com/qhosting/escalafin-mvp/commits/main>



## Solución de Problemas

### Error: "fatal: could not read Username"

**Solución:** Usa un Personal Access Token (PAT) como se explicó arriba.

### Error: "Authentication failed"

**Solución:** Verifica que estás usando el token correcto (no tu password).

## Error: “Permission denied”

**Solución:** Asegúrate de tener permisos de escritura en el repositorio.

## Error: “Updates were rejected”

**Solución:** Primero haz pull y luego push:

```
git pull origin main --rebase
git push origin main
```



## Contenido de los Commits Pendientes

Los commits que se van a subir incluyen:

### 1. Multi-instance deployment & local testing

- Infraestructura de despliegue multi-instancia
- Scripts de Coolify
- Configuración de testing local

### 2. Production deployment tested and verified

- Verificación completa de build de producción
- Reportes de deployment

### 3. Multi-instance Coolify deployment ready

- Templates de Coolify
- Scripts de automatización
- Guías de despliegue

### 4. Variables de entorno para EasyPanel

- Documentación de configuración
- Guías de setup



## Próximos Pasos Después del Push

Una vez que los cambios estén en GitHub:

### 1. Desplegar en producción:

- Conecta tu servicio de hosting (Vercel, Railway, etc.) al repo
- O usa los scripts de Coolify incluidos

### 2. Configurar CI/CD:

- GitHub Actions ya está configurado
- Los deploys automáticos se activarán

### 3. Revisar el deployment:

- Verifica que todo funcione correctamente
  - Revisa los logs de build
-

## Soporte

---

Si tienes problemas:

1. Revisa la sección de solución de problemas
2. Verifica los logs: `git --version` y `git config --list`
3. Asegúrate de tener conexión a internet

**Repositorio:** <https://github.com/qhosting/escalafin-mvp>

**Commits pendientes:** 4

¡Éxito con tu despliegue! 🚀