

Guía Completa para Colaborar en el Repositorio GitHub

Esta guía explica **cómo trabajar correctamente en el repositorio**, cómo el administrador debe agregar colaboradores, cómo generar **Tokens Personales (PAT)** para poder hacer `git push`, y el proceso recomendado de trabajo usando **ramas y Pull Requests**.

1. Para el Administrador: Agregar Colaboradores al Repositorio

El administrador debe dar acceso a cada miembro del equipo.

Pasos:

1. Entrar al repositorio en GitHub.
2. Ir a **Settings**.
3. En el menú izquierdo, seleccionar **Collaborators & Teams**.
4. Clic en **Add Collaborator**.
5. Escribir el nombre del usuario de GitHub (ej. `JonSDRod`).
6. Seleccionar permiso **Write**.
7. Enviar invitación.

El colaborador debe **aceptar la invitación** en: - Su correo electrónico, o - <https://github.com/notifications>

Si no la acepta, GitHub no permitirá hacer push y mostrará error 403.

2. Crear un Token Personal (PAT) para hacer Push

GitHub ya **NO permite contraseñas** en Git. Solo acepta: - Tokens personales - Llaves SSH

El método más sencillo es usar **Personal Access Token (Classic)**.

2.1 Crear un Token Personal

1. Ir a GitHub.
2. Clic en tu foto de perfil (arriba a la derecha).
3. **Settings**.
4. Abajo del menú izquierdo → **Developer settings**.

5. **Personal access tokens** → **Tokens (classic)**.
6. **Generate new token** → "Generate new token (classic)".
7. Ponle un nombre (ej. "token para pushes").
8. Elegir permisos:
9. ☐ repo ☒ (necesario)
10. Generar token.

IMPORTANTE: GitHub solo muestra el token una vez. Cópiarlo y guárdalo.

2.2 Usar el Token en el Push

Cuando hagas:

```
git push origin rama-nombre
```

Git pedirá:

```
Username: TU_USUARIO
Password: <AQUÍ PEGAS EL TOKEN>
```

Tu contraseña real ya no sirve. El token es tu nueva "contraseña".

2.3 Guardar el token para no escribirlo cada vez

```
git config --global credential.helper store
```

Después del primer push con token, queda guardado.

3. Flujo de Trabajo para Todos los Integrantes

Usaremos ramas y Pull Requests para que cada integrante pueda subir su archivo y que el administrador lo apruebe.

3.1 Clonar el repositorio

```
git clone https://github.com/TU_ORGANIZACION/TU_REPO.git  
cd TU_REPO
```

3.2 Crear una rama personal

Cada integrante debe trabajar en su propia rama.

```
git checkout -b rama-nombre
```

Ejemplo:

```
git checkout -b rama-juan
```

3.3 Crear su archivo

```
touch archivo_nombre.py
```

Editar el archivo normalmente.

3.4 Agregar y hacer commit

```
git add archivo_nombre.py  
git commit -m "Agrego archivo de Nombre"
```

3.5 Subir la rama al repositorio remoto

```
git push origin rama-nombre
```

Si pide contraseña, pega tu **Token Personal**.

4. Crear un Pull Request (PR)

En GitHub: 1. Entrar al repositorio. 2. GitHub mostrará **Compare & Pull Request**. 3. Hacer clic. 4. Describir qué se subió. 5. Enviar Pull Request.

5. Para el Administrador: Revisar y Aceptar PRs

1. Entrar al repositorio.
2. Ir a **Pull Requests**.
3. Abrir el PR correspondiente.
4. Revisar cambios.
5. Si está correcto → **Merge Pull Request**.
6. Si falta algo → agregar comentarios.

Así el archivo se integra a `main`.

6. Actualizar el repositorio local de todos

Después de que el administrador aceptó un PR:

```
git checkout main
git pull origin main
```

7. Notas Importantes

- No trabajar directamente en la rama `main`.
 - Cada integrante debe aceptar la invitación de colaborador.
 - Si aparece error 403, significa falta de permisos.
 - Si aparece "password not supported", falta token o SSH.
-

Fin del documento.