**🚀 Flujo completo para próximos proyectos en Replit**

**1. Crear el proyecto desde un template**

* En Replit, pulsa **Remix** en el template → se crea un nuevo proyecto con todos los archivos.
* Abre la shell en ese proyecto.

**2. Eliminar el .git heredado del template**

Así evitas que se convierta en submódulo.

rm -rf .git

**3. Conectar tu repo de GitHub**

Clona tu repo en el workspace del proyecto remixado:

git clone https://github.com/MMudeLL/Cursor-Replit-apps.git

**4. Crear carpeta para tu nueva app**

mkdir Cursor-Replit-apps/basic-chatbot

*(Cambia basic-chatbot por el nombre de tu app nueva)*

**5. Mover todos los archivos del template a esa carpeta**

mv \* .[^.]\* Cursor-Replit-apps/basic-chatbot 2>/dev/null || true

**6. Subir al repo**

Entra a tu repo clonado y sube los cambios:

cd Cursor-Replit-apps

git add basic-chatbot

git commit -m "Añadido template basic-chatbot"

git push origin main

**7. 🚑 Si aparece el error de submódulo (flecha ➡️ en GitHub)**

Corre estos pasos para corregirlo:

git rm -r --cached basic-chatbot

rm -rf basic-chatbot/.git

git add basic-chatbot

git commit -m "Corrijo basic-chatbot como carpeta normal"

git push origin main

✅ Con eso, cada template que remixes quedará limpio, dentro de su carpeta en tu repo principal, y sin submódulos.

¡Genial! 🙌 Le añadimos el **fix automático de submódulos** para que no tengas que preocuparte más por la flechita ➡️ en GitHub.

**📄 Script final subir-template.sh**

#!/bin/bash

# Uso: ./subir-template.sh nombre-app

APP\_NAME=$1

if [ -z "$APP\_NAME" ]; then

echo "❌ Debes indicar un nombre de app. Ejemplo: ./subir-template.sh basic-chatbot"

exit 1

fi

# 1. Borrar .git heredado del template remixado

echo "🧹 Eliminando .git del template (si existe)..."

rm -rf .git

# 2. Clonar tu repositorio si no existe

if [ ! -d "Cursor-Replit-apps" ]; then

echo "📥 Clonando repo Cursor-Replit-apps..."

git clone https://github.com/MMudeLL/Cursor-Replit-apps.git

fi

# 3. Crear carpeta para la app

echo "📂 Creando carpeta Cursor-Replit-apps/$APP\_NAME..."

mkdir -p Cursor-Replit-apps/$APP\_NAME

# 4. Mover archivos del template a la carpeta

echo "📦 Moviendo archivos del template a la carpeta..."

mv \* .[^.]\* Cursor-Replit-apps/$APP\_NAME 2>/dev/null || true

# 5. Entrar en el repo

cd Cursor-Replit-apps || exit

# 6. Fix automático de submódulos (por si hay .git internos)

if [ -d "$APP\_NAME/.git" ]; then

echo "⚠️ Se detectó un .git interno en $APP\_NAME. Corrigiendo..."

git rm -r --cached $APP\_NAME

rm -rf $APP\_NAME/.git

fi

# 7. Añadir, commitear y subir

echo "✅ Añadiendo archivos al repo..."

git add $APP\_NAME

git commit -m "Añadido template $APP\_NAME"

git push origin main

echo "🎉 Listo! Tu app '$APP\_NAME' ya está en GitHub dentro de Cursor-Replit-apps/$APP\_NAME"

**🔹 Cómo usarlo**

1. Crear el archivo en tu proyecto remixado:

nano subir-template.sh

1. Pegar el script completo.
2. Guardar y dar permisos de ejecución:

chmod +x subir-template.sh

1. Ejecutar con el nombre de la app:

./subir-template.sh basic-chatbot

**🔹 Qué hace**

* Borra .git heredado del template.
* Clona tu repo Cursor-Replit-apps si no está.
* Crea la carpeta para la app.
* Mueve todos los archivos allí.
* Si hay .git dentro → lo elimina y evita que sea submódulo.
* Hace commit y push automáticamente.