# Reemplaza la función 'ejecutar\_calculo\_externo' en dem25\_tab.py con esta

import sys # Asegúrate de que 'import sys' está al principio de tu script

def ejecutar\_calculo\_externo(dem\_bytes, outlet\_coords\_wgs84, umbral):

"""

Ejecuta el script 'delinear\_cuenca.py' en su propio entorno virtual

de forma adaptable para Windows y Linux.

"""

# --- 1. Detección del Sistema Operativo y Configuración de Rutas ---

# Determinar el nombre de la carpeta del ejecutable ('Scripts' o 'bin')

if sys.platform == "win32":

# Estamos en Windows

bin\_folder = "Scripts"

python\_exe = "python.exe"

else:

# Asumimos Linux o macOS

bin\_folder = "bin"

python\_exe = "python"

# Construir las rutas de forma dinámica

base\_path = "Py2Env"

venv\_path = os.path.join(base\_path, "venv\_pysheds")

python\_executable\_externo = os.path.join(venv\_path, bin\_folder, python\_exe)

script\_path = os.path.join(base\_path, "delinear\_cuenca.py")

# Verificación de existencia de las rutas

if not os.path.exists(python\_executable\_externo):

st.error(f"Error Crítico: No se encuentra el ejecutable de Python en la ruta esperada: '{python\_executable\_externo}'.")

return None

if not os.path.exists(script\_path):

st.error(f"Error Crítico: No se encuentra el script en la ruta: '{script\_path}'.")

return None

# --- 2. Transformación de Coordenadas (sin cambios) ---

try:

transformer\_wgs84\_to\_utm30n = Transformer.from\_crs("EPSG:4326", "EPSG:25830", always\_xy=True)

x\_utm, y\_utm = transformer\_wgs84\_to\_utm30n.transform(outlet\_coords\_wgs84['lng'], outlet\_coords\_wgs84['lat'])

outlet\_coords\_utm = {"x": x\_utm, "y": y\_utm}

except Exception as e:

st.error(f"Error al transformar coordenadas: {e}")

return None

# --- 3. Ejecución del Subproceso (sin cambios) ---

with tempfile.TemporaryDirectory() as tmpdir:

dem\_input\_path = os.path.join(tmpdir, "input\_dem.tif")

with open(dem\_input\_path, 'wb') as f:

f.write(dem\_bytes)

comando = [

python\_executable\_externo,

script\_path,

dem\_input\_path,

json.dumps(outlet\_coords\_utm),

str(umbral)

]

try:

with st.spinner("Ejecutando análisis hidrológico completo..."):

resultado = subprocess.run(

comando, capture\_output=True, text=True, check=True, encoding='utf-8'

)

return json.loads(resultado.stdout)

except subprocess.CalledProcessError as e:

st.error("Falló la ejecución del script externo. Detalles del error:")

st.code(e.stderr, language='bash')

return None

except Exception as e:

st.error(f"Ocurrió un error inesperado al llamar al subproceso: {e}")

return None