



---

# PROYECTO FINAL

---

MAPA DEL ECUADOR 3D



**ASIGNATURA:**  
METODOS NUMERICOS

**INTEGRANTES:**  
LADY VELASQUEZ  
JOSUE CALVOPÍÑA MACHUCA  
MAURICIO CARRION  
KEVIN GARCIA  
LUIS LEMA

**FECHA DE ENTREGA:**  
02/02/2026

## Contenido

<b>OBJETIVOS.....</b>	<b>1</b>
<b>MARCO TEÓRICO .....</b>	<b>1</b>
<b>PRERREQUISITOS .....</b>	<b>¡Error! Marcador no definido.</b>
<b>DESARROLLO.....</b>	<b>6</b>
<b>CONCLUSIONES.....</b>	<b>9</b>
<b>RECOMENDACIONES .....</b>	<b>9</b>
<b>REFERENCIAS .....</b>	<b>9</b>

## OBJETIVOS

- Crear un objeto para impresión 3D con escala realista en un área determinada.

## MARCO TEÓRICO

Para la transformación de datos crudos en modelos físicos, el proyecto se fundamenta en los siguientes conceptos:

### Modelos Digitales de Elevación (DEM)

Un DEM es una representación visual y matemática de los valores de altura del terreno en relación con el nivel medio del mar. En este proyecto se utilizan datos de la misión SRTM (Shuttle Radar Topography Mission), estructurados como matrices numéricas donde cada celda representa la altura en metros.

### Procesamiento Vectorial y Raster

El sistema integra dos tipos de datos geoespaciales:

- Raster (.hgt/.tif): Matrices de píxeles que contienen información de altura
- Vectorial (.json/GeoJSON): Polígonos geométricos que definen los límites administrativos de las provincias. El uso de máscaras vectoriales permite recortar el raster siguiendo formas irregulares exactas, superior al recorte rectangular tradicional.

### Algoritmo de Filtrado Numérico

Para el tratamiento de ruido en la matriz de elevación, se aplican dos filtros espaciales:

- Filtro de Promedio: Técnica lineal que calcula la media aritmética de una ventana deslizante (3x3). Es eficaz para suavizar superficies rugosas, aunque tiene a erosionar bordes agudos
- Filtro de Mediana: Técnica estadística no lineal que reemplaza el valor del píxel por la mediana de su vecindad. Este método es superior teóricamente para eliminar “ruido impulsivo” que son errores puntuales de satélite preservando la morfología de accidentes geográficos como cráteres y acantilados.

### Formato STL

Para que una impresora 3D pueda materializar un objeto, necesita una definición geométrica precisa de su superficie. El estándar utilizado en la industria es el formato STL (Standard Triangle Language).

Este formato funciona mediante un proceso conocido como teselación o triangulación. Cualquier superficie curva o irregular (como una montaña) se aproxima mediante miles de pequeños triángulos planos unidos entre sí.

### Watertight

Un concepto fundamental en la impresión 3D es la diferencia entre una "superficie" y un "sólido".

- **Superficie:** Es una capa sin grosor (como una hoja de papel teórica). Las impresoras no pueden imprimir esto porque no tiene volumen físico.
- **Sólido (Watertight):** Es un objeto cerrado herméticamente.

Para que el modelo topográfico sea funcional, no basta con tener la malla del terreno; es obligatorio que el objeto sea "manifold". Esto significa que la malla debe tener paredes laterales y una base plana

que cierren el volumen, permitiendo al software de impresión, calcular el relleno y las capas de plástico correctamente.

## Técnicas de Suavizado

### Filtro Promedio o media

El filtro de promedio reemplaza cada píxel por el promedio de los valores de sus vecinos dentro de una ventana. Esto suaviza la imagen reduciendo variaciones bruscas.

$$I'(x, y) = \frac{1}{m \cdot n} \sum_{i=-a}^a \sum_{j=-b}^b I(x + i, y + j)$$

- $I(x,y)$ : valor original del píxel.
- $I'(x,y)$ : valor filtrado.
- La suma recorre todos los vecinos dentro de la ventana centrada en  $(x,y)$ .
- El factor  $\frac{1}{m \cdot n}$  asegura que el resultado sea un promedio.

Si el kernel es de tamaño  $m \times n$ :

- $a = \frac{(m-1)}{2}$  → número de pasos hacia arriba/abajo en la dirección **x**.
- $b = \frac{(n-1)}{2}$  → número de pasos hacia izquierda/derecha en la dirección **y**.

Reduce ruido y Borra detalles y bordes

En nuestro proyecto lo aplican 3 veces:

*matriz\_suave = suavizado\_promedio(matriz)*

Aquí está la matemática del suavizado dentro del código usando la librería `scipy.ndimage`.

*`uniform_filter` → filtro promedio*

*`median_filter` → filtro mediana*

Esto termina siendo un procesamiento digital de señales / imágenes.



## Filtro Mediana

El filtro de mediana es un filtro no lineal que reemplaza el valor de cada píxel por la mediana de los valores en su vecindario. A diferencia del filtro de promedio, que suaviza difuminando, el de mediana preserva mejor los bordes y elimina ruido impulsivo.

Sea una imagen  $I(x,y)$  y un kernel de tamaño  $m \times n$ . El valor filtrado en la posición  $(x,y)$  se calcula como:

$$I'(x,y) = \text{mediana}(\{I(x+i, y+j) \mid i \in [-a, a], j \in [-b, b]\})$$

- $a = \frac{(m-1)}{2}$
- $b = \frac{(n-1)}{2}$
- El conjunto dentro de la mediana contiene los valores de todos los píxeles vecinos en la ventana centrada en  $(x,y)$ .
- La mediana es el valor central cuando se ordenan todos esos valores de menor a mayor.

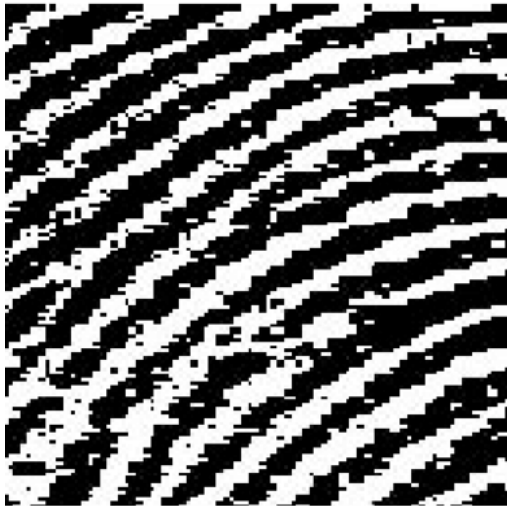
Supongamos que el vecindario de un píxel es:

*[10 200 15 20 25 30 22 18 250]*

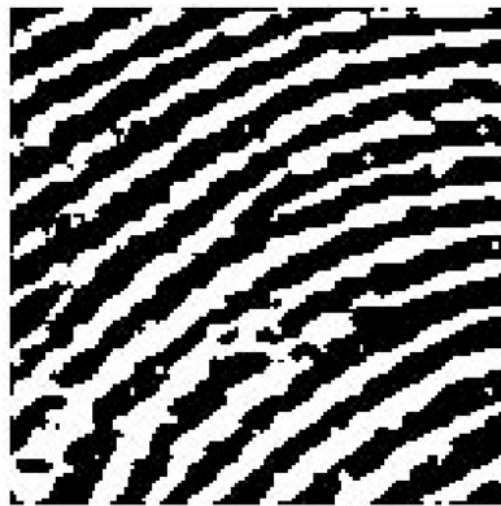
- Paso 1: Tomamos todos los valores  $\rightarrow \{10,200,15,20,25,30,22,18,250\}$ .
- Paso 2: Ordenamos  $\rightarrow \{10,15,18,20,22,25,30,200,250\}$ .
- Paso 3: La mediana es el valor central  $\rightarrow 22$ .

Entonces el píxel filtrado  $I'(x,y)=22$ .

- El filtro de mediana es muy usado en procesamiento de imágenes para limpiar ruido tipo “sal y pimienta”.
- Matemáticamente, en vez de calcular un promedio, se busca el valor más representativo del vecindario.
- Por eso preserva mejor los bordes: no se dejan arrastrar por valores extremos.
- **Lineal:** operaciones basadas en suma y multiplicación (como el promedio).
- **No lineal:** operaciones que dependen de orden, comparación o selección (como la mediana, el máximo, el mínimo).



Original



Filtrada (Mediana)

Aspecto	Filtro de Promedio	Filtro de Mediana
<b>Tipo</b>	Lineal	No lineal
<b>Operación</b>	Suma y división	Ordenar y tomar valor central
<b>Efecto</b>	Suaviza todo, difumina bordes	Elimina ruido impulsivo, preserva bordes
<b>Ejemplo</b>	Ruido se difumina	Ruido se elimina directamente

Como definición extra tenemos que:

*El ruido de sal y pimienta es un tipo de interferencia en imágenes digitales que se manifiesta como píxeles blancos (sal) y negros (pimienta) distribuidos aleatoriamente, degradando la calidad visual. Generalmente causado por fallos en el sensor de la cámara, errores de transmisión o almacenamiento, este ruido "impulsivo" satura los valores de píxel al máximo (blanco) o al mínimo (negro). Es un defecto común que se puede corregir eficazmente mediante filtros de mediana.*

## PRERREQUISITO

Para el funcionamiento del sistema desarrollado, se requiere una configuración previa del entorno de software y datos.

### Gestión de Datos

Debido al gran volumen de los archivos de topografía y modelos 3D, el repositorio utiliza Git Large File Storage (LFS), para el comando de instalación "git lfs install", el comando de descarga de datos binarios "git lfs pull"

### Estructura de Datos

El sistema requiere dos fuentes de información en el directorio raíz:

- Datos Raster (datos\_srtm): Son los archivos .hgt con la elevación del terreno.
- Datos Vectoriales (gadm41\_ECU\_1.json): Archivo GeoJSON con los polígonos administrativos de nivel 1 (Provincias) de Ecuador.

### Librerías

El código al ser desarrollado en python, depende de las siguientes librerías.

Librería	Función en el Proyecto
geopandas	Carga y manipulación de los polígonos de provincias. Permite operaciones geométricas complejas.
rasterio	Lectura de datos satelitales, fusión de mapas y aplicación de máscaras de recorte (mask).
numpy	Procesamiento matricial de alto rendimiento para los cálculos de elevación.
scipy	Implementación optimizada de los algoritmos de suavizado numérico (ndimage).
numpy-stl	Generación de la malla triangular 3D y exportación del archivo final.
matplotlib	Interfaz gráfica de usuario (GUI) para la selección interactiva y visualización.

## DESARROLLO

### Integración de datos SRTM (.hgt) en la generación de Modelos Digitales de Elevación

Para la transformación de datos crudos en modelos físicos del relieve, el proyecto se fundamenta en el uso de archivos .hgt generados por la misión SRTM (Shuttle Radar Topography Mission), los cuales constituyen Modelos Digitales de Elevación (DEM). Un DEM es una representación visual y matemática de los valores de altura del terreno en relación con el nivel medio del mar, donde cada celda de una matriz regular almacena un valor altimétrico expresado en metros. Estos archivos no corresponden a imágenes convencionales, sino a estructuras numéricas sin compresión, organizadas como grillas de datos continuos que deben ser interpretadas mediante librerías geoespaciales, permitiendo transformar una secuencia de números en una superficie topográfica comprensible.

Cada archivo .hgt representa únicamente un fragmento del territorio, equivalente a un grado de latitud por un grado de longitud, por lo que cualquier región real de estudio se encuentra inevitablemente distribuida en múltiples bloques independientes. Esta fragmentación hace necesario aplicar un proceso de mosaico raster, mediante el cual se integran todos los archivos requeridos en un solo modelo continuo. La unión de estos fragmentos permite reconstruir el relieve de manera coherente, eliminando discontinuidades entre bordes y garantizando la continuidad espacial de la superficie, de modo que el modelo final conserve la relación geográfica real entre todos sus puntos.





Dentro de este sistema se integran dos tipos de datos geoespaciales:

- Raster (.hgt / .tif): Matrices de píxeles que contienen información numérica de elevación. Cada píxel representa la altura del terreno en un punto específico, conformando la base matemática del Modelo Digital de Elevación.
- Vectorial (.json / GeoJSON): Polígonos geométricos que delimitan unidades administrativas o áreas de estudio. Estas capas vectoriales se utilizan como máscaras para recortar el raster siguiendo formas irregulares reales, lo cual resulta superior al recorte rectangular tradicional, ya que permite conservar únicamente el territorio de interés.

La integración entre datos raster y vectoriales permite generar un modelo de elevación continuo, preciso y territorialmente delimitado. De esta manera, el DEM resultante no solo representa la forma del terreno, sino que se convierte en una herramienta analítica para el estudio del relieve, la planificación territorial y la interpretación de procesos naturales, consolidando a los archivos .hgt como un elemento esencial en la construcción de modelos geoespaciales confiables.

### **Limitación por provincias:**

La limitación por provincias en este proyecto es un sistema de segmentación geográfica que permite generar modelos 3D STL de regiones específicas de Ecuador usando los límites administrativos.

El proyecto utiliza el archivo GeoJSON que contiene polígonos vectoriales de las provincias ecuatorianas del proyecto GADM (Global Administrative Areas). Este archivo define los límites exactos de cada provincia mediante coordenadas geográficas.

La carga y procesamiento de del GeoJSON se realiza dentro de la función “cargar\_provincias” se lee con GeoPandas y normaliza los nombres de columnas (provincias), este retorna un GeoDataFrame con geometría (polígonos) de cada provincia.

La función “generar\_stl\_provincia()” implementa el algoritmo central de extracción geográfica mediante un proceso de dos etapas fundamentales:

**Reproyección de Coordenadas:** El sistema primero verifica que el polígono de la provincia y el raster de elevación utilicen el mismo Sistema de Referencia de Coordenadas (CRS). Si no coinciden, realiza automáticamente una transformación de coordenadas para alinear ambos sistemas espaciales. Esto es esencial porque trabajar con sistemas de coordenadas diferentes generaría desalineaciones que afectarían la precisión del recorte.

**Masking y Recorte:** Una vez alineados los sistemas de coordenadas, el algoritmo aplica el polígono provincial como una máscara sobre el raster completo de elevación de Ecuador. Esta operación utiliza la función “rasterio\_mask()” que identifica qué píxeles del raster caen dentro del polígono y cuáles están fuera. Los píxeles exteriores reciben un valor especial de "sin datos" (-32768), mientras que los interiores mantienen sus valores de elevación originales. El parámetro “crop=True” adicional hace que el sistema recorte el raster al extent mínimo rectangular que contiene completamente la provincia, optimizando el uso de memoria y mejorando el rendimiento.

El sistema implementa un mecanismo robusto de verificación que genera automáticamente imágenes de control guardadas en la carpeta “verificacion\_provincias”. Cada imagen muestra la matriz de

elevación recortada utilizando un mapa de colores tipo "terrain" que facilita la interpretación visual del relieve.

Sobre esta visualización, el algoritmo superpone el contorno exacto del polígono provincial. Para lograr esto, transforma las coordenadas geográficas del polígono a índices de píxeles dentro del raster recortado, utilizando la transformación afín inversa. Esto permite verificar visualmente que el recorte se realizó correctamente y que los límites coinciden con las fronteras administrativas esperadas.

Cada imagen incluye metadatos importantes como el área en píxeles (dimensiones de la matriz), y el rango de elevación mínimo y máximo detectado en esa provincia específica.

## Generación del STL

Esta etapa del proyecto se encarga de transformar la matriz numérica de elevaciones procesada anteriormente en un archivo físico .stl. El proceso se ejecuta mediante un algoritmo desarrollado en Python (módulo procesador.py) que utiliza la librería numpy-stl para el cálculo vectorial. El procedimiento sigue tres fases:

### 1. Transformación de Espacio y Escalado

Los datos de entrada consisten en una matriz de dimensiones  $M \times N$  donde cada celda  $(i, j)$  contiene un valor de elevación en metros. Dado que el espacio de impresión 3D es limitado y métrico (milímetros), se aplicó una transformación lineal para mapear el espacio geográfico al espacio de la impresora.

Se define un factor de escala global ( $S$ ) basado en la dimensión máxima deseada para la impresión ( $L_{max}$ ):

$$S = L_{max} / \max(M, N)$$

Posteriormente, cada vértice  $V$  del modelo se calcula transformando sus índices  $(i, j)$  y su altura  $Z$  mediante las ecuaciones de escalado, donde  $E_z$  es el factor de exageración vertical:

$$\begin{aligned}x &= j \cdot S \\y &= i \cdot S \\z &= Z(i, j) \cdot S \cdot E_z\end{aligned}$$

### 2. Algoritmo de Teselación (Triangulación Vectorizada)

Para generar la superficie continua, se utilizó un método de triangulación regular. El algoritmo recorre la grilla de puntos y divide cada celda cuadrangular definida por los vértices  $(i, j)$  en dos triángulos planos ( $T1$  y  $T2$ ).

La topología de la malla se define conectando los vértices en sentido antihorario para asegurar la correcta orientación de las caras. Matemáticamente, para cada posición  $(i, j)$ , los triángulos se definen por los siguientes conjuntos de vértices:

$$\text{Triángulo 1 (T1): } \{ V(i, j), \quad V(i, j + 1), \quad V(i + 1, j) \}$$

$$\text{Triángulo 2 (T2): } \{ V(i + 1, j + 1), \quad V(i + 1, j), \quad V(i, j + 1) \}$$

Este proceso se realiza de forma vectorizada utilizando la librería numpy, lo que permite generar millones de triángulos simultáneamente sin recurrir a bucles iterativos, optimizando el rendimiento del procesador.

#### 4.3. Cierre de Volumen y Cálculo de Normales

Para cumplir con la condición de “watertight” requerida para la impresión, se generaron superficies adicionales para los bordes y la base.

La orientación de cada triángulo es crítica y está determinada por su vector normal ( $\mathbf{n}$ ). El algoritmo calcula la normal de cada cara mediante el producto cruz de dos de sus vectores arista ( $\mathbf{U}$  y  $\mathbf{V}$ ):

$$\mathbf{n} = \mathbf{U} \times \mathbf{V}$$

Donde los vectores arista se definen como la diferencia entre sus vértices:

$$\mathbf{U} = \mathbf{V2} - \mathbf{V1}$$

$$\mathbf{V} = \mathbf{V3} - \mathbf{V1}$$

El código asegura que todos los vectores normales  $\mathbf{n}$  apunten hacia el exterior del volumen. Finalmente, se fusionan la malla del terreno, las paredes laterales (generadas por extrusión de los bordes hacia  $\mathbf{z} = 0$ ) y la base inferior, consolidando un único objeto geométrico cerrado listo para la exportación.

## CONCLUSIONES

- Se logró automatizar exitosamente el flujo de trabajo cartográfico, permitiendo la generación de modelos 3D precisos tanto por selección manual como por límites provinciales exactos mediante el uso de **GeoPandas**.
- La comparación de algoritmos demostró que el **Filtro de Mediana** es superior para aplicaciones topográficas volcánicas, ya que elimina los errores de medición del sensor SRTM sin degradar la pendiente de los cráteres, a diferencia del promedio que tiende a suavizar en exceso.

## RECOMENDACIONES

## REFERENCIAS

[1] J. E. Sanchez Tarquino, "Estudio de técnicas de suavización de imágenes basadas en modelos dispersos," Tesis de Grado, Facultad de Ingeniería, Universidad de Ibagué, Ibagué, Colombia, 2019.