



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

## **HERRAMIENTA INTERACTIVA PARA VISUALIZAR Y OPERAR SOBRE MALLAS DE TETRAEDROS**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

SEBASTIÁN ANTONIO CORTÉS CATALÁN

PROFESORA GUÍA:  
MARÍA CECILIA RIVARA ZÚÑIGA

MIEMBROS DE LA COMISIÓN:  
LUIS MATEU BRULE  
GONZALO NAVARRO BADINO

SANTIAGO DE CHILE  
2024

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN  
POR: SEBASTIÁN ANTONIO CORTÉS CATALÁN  
FECHA: 2024  
PROF. GUÍA: María Cecilia Rivara Zúñiga

## **HERRAMIENTA INTERACTIVA PARA VISUALIZAR Y OPERAR SOBRE MALLAS DE TETRAEDROS**

En este trabajo se presenta un visualizador interactivo de mallas de tetraedros, herramienta que permite la visualización y manipulación de mallas volumétricas dentro de un espacio tridimensional. El proyecto está centrado principalmente en la representación de la malla y operaciones sobre tetraedros, que son elementos fundamentales para la simulación numérica y la representación tridimensional en diversas áreas científicas y de ingeniería.

La herramienta posee distintas funcionalidades, como la inserción de puntos, el suavizado de superficie y el intercambio de aristas, las cuales juegan un papel crucial en obtener resultados precisos y eficientes en diversas aplicaciones científicas e industriales.

Para lo anterior, se utiliza una estructura de datos, la cual usa una representación simple de la malla, utilizando sus vértices, aristas, tetraedros y sus vecinos. Esta estructura permite tener un acceso eficiente a la información necesaria para la visualización y ejecución de operaciones. Además, se utilizará solo el formato de archivo .m3d para la carga de datos.

La visualización de los tetraedros es realizada mediante formas sólidas en el espacio 3D, con colores para resaltar características específicas. Los usuarios pueden interactuar con la visualización, rotando, moviendo y ajustando el tamaño de los tetraedros para explorar la estructura interna y externa de la malla.

La herramienta implementa algoritmos LEPP para su uso en las mallas tetraédricas y ofrece una serie de operaciones que se pueden realizar sobre la malla de tetraedros, como la inserción de puntos, el suavizado de superficie y el intercambio de aristas. Estas operaciones se aplican a los tetraedros seleccionados por el usuario, proporcionando una experiencia interactiva y amigable.

Finalmente, este trabajo ofrece una manera eficiente y efectiva para la visualización y operaciones sobre mallas de tetraedros. La aplicación tiene gran utilidad como una herramienta didáctica para aprender conceptos sobre estas mallas.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
<b>2. Conceptos Previos</b>	<b>3</b>
2.1. Mallas volumétricas . . . . .	3
2.2. Triangulación de Delaunay . . . . .	3
2.3. Arista terminal . . . . .	4
2.4. Estrella terminal . . . . .	4
2.5. Algoritmos LEPP . . . . .	5
2.6. Operaciones . . . . .	6
2.6.1. Inserción de punto . . . . .	6
2.6.2. Suavizamiento . . . . .	6
2.6.3. Intercambio de aristas . . . . .	6
2.7. Motor de videojuego . . . . .	6
<b>3. Fundamentación del tema</b>	<b>8</b>
3.1. Avances del área . . . . .	8
3.2. Visualizadores de mallas . . . . .	8
3.3. Herramientas didácticas . . . . .	10
<b>4. Diseño de la aplicación</b>	<b>11</b>
4.1. Estructura de datos . . . . .	11
4.2. Manejo de archivos . . . . .	12
4.3. Visualización de tetraedros . . . . .	13
4.4. Algoritmo LEPP . . . . .	13
4.4.1. LEPP simple . . . . .	13
4.4.2. LEPP completo . . . . .	14
4.5. Operaciones . . . . .	15
4.5.1. Inserción de punto . . . . .	15
4.5.2. Suavizamiento . . . . .	15
4.5.3. Intercambio de aristas . . . . .	16
4.6. Interfaz Gráfica . . . . .	16
<b>5. Resultados</b>	<b>18</b>
<b>6. Evaluación</b>	<b>25</b>
6.1. Rendimiento . . . . .	25
6.2. Validación . . . . .	29

<b>7. Conclusiones</b>	<b>34</b>
7.1. Limitaciones . . . . .	34
7.2. Sugerencias y futuros trabajos . . . . .	35
<b>Bibliografía</b>	<b>36</b>

# Índice de Ilustraciones

1.1.	Malla geométrica 2D. . . . .	1
1.2.	Malla de tetraedros de un eje [1]. . . . .	1
3.1.	MeshLab. . . . .	9
4.1.	Representación de tetraedro. . . . .	11
5.1.	Pantalla inicial. . . . .	18
5.2.	Menú de selección. . . . .	19
5.3.	Visualización. . . . .	19
5.4.	Visualización con menú. . . . .	20
5.5.	Menú de pausa. . . . .	20
5.6.	Visualización con tamaño modificado. . . . .	21
5.7.	Selección de tetraedro. . . . .	21
5.8.	Vecinos de un tetraedro. . . . .	22
5.9.	Algoritmo LEPP simple. . . . .	22
5.10.	Algoritmo LEPP completo. . . . .	23
5.11.	Inserción de centroide. . . . .	23
5.12.	Suavizamiento. . . . .	23
5.13.	Intercambio de arista. . . . .	24
5.14.	Vecinos de un conjunto de tetraedros. . . . .	24
6.1.	Rendimiento en función de la cantidad de vértices del modelo. . . . .	26
6.2.	Rendimiento en función de la cantidad de tetraedros del modelo. . . . .	26
6.3.	Tiempo de cada fotograma en función de la cantidad de vértices. . . . .	27
6.4.	Tiempo de cada fotograma en función de la cantidad de tetraedros. . . . .	27
6.5.	Rendimiento del algoritmo simple LEPP. . . . .	28
6.6.	Rendimiento de la operación de inserción. . . . .	28
6.7.	Rendimiento de la operación de suavizamiento. . . . .	28
6.8.	Vecinos del tetraedro original de la malla. . . . .	29
6.9.	vecinos del tetraedro nuevo de la malla modificada. . . . .	30
6.10.	Validación de los primero 30 tetraedros de la malla original. . . . .	30
6.11.	Validación de la malla original. . . . .	31
6.12.	Inserción de centroide en la malla. . . . .	32
6.13.	Validación de los primeros 30 tetraedros de la malla modificada. . . . .	33
6.14.	Validación de la malla modificada. . . . .	33

# Capítulo 1

## Introducción

Una malla geométrica es un instrumento de gran utilidad en distintos ámbitos como, por ejemplo, en aplicaciones del método de elementos finitos (MEF), modelamiento de fenómenos físicos y modelamientos tridimensionales. Estas mallas pueden ser de 2 o 3 dimensiones. Las mallas en 2D pueden ser utilizadas para modelar planos o superficies (Figura 1.1) y las de 3 dimensiones se pueden usar para discretizar y modelar tanto la parte interna como la parte externa de un objeto (Figura 1.2).

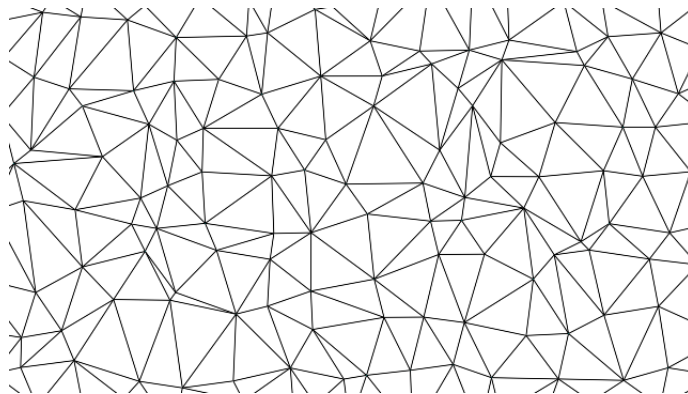


Figura 1.1: Malla geométrica 2D.

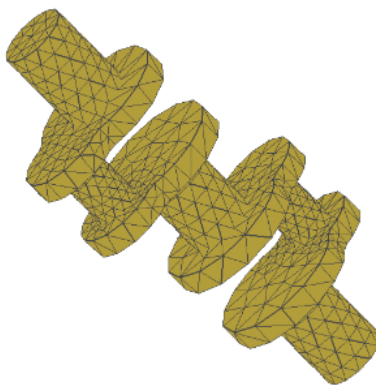


Figura 1.2: Malla de tetraedros de un eje [1].

Este proyecto trabaja con mallas de tetraedros, las cuales se componen de tetraedros

unidos por sus vértices, aristas o caras, pero que no se superponen entre sí, logrando así formar una aproximación del dominio que se quiere representar.

Teniendo esto en cuenta, el poder visualizar e interactuar con este tipo de mallas es algo de gran interés, tanto para investigadores como para estudiantes del área. Sin embargo, no existen muchas herramientas para poder visualizar mallas en 3D y las que existen no tienen como objetivo mostrar las operaciones que se ejercen sobre éstas. Además, estos programas son más complejos de usar, lo que restringe su uso y dificulta el crecimiento de esta área.

Estas mallas pueden ser objeto de distintos tipos de operaciones, como lo son el intercambio de aristas y la inserción de vértices en estrellas terminales [2]. Las visualizaciones de estas operaciones no existen en los programas de hoy. Por demás, dada la naturaleza en 3 dimensiones de este tipo de mallas, tener una idea visual del comportamiento que ésta tendrá resulta algo difícil, más aún en mallas complejas.

Dada la complejidad que poseen estas operaciones en este contexto, se optará por restringir el trabajo a operaciones no Delaunay. Una operación no Delaunay es una que no preservan la condición de una triangulación de Delaunay. Esta condición se logra cuando las esferas circunscritas de cada tetraedro no encierran o tocan a ningún vértice perteneciente a otro tetraedro [3]. Se opta por este tipo de operaciones debido a la complejidad que tiene cumplir con la condición de Delaunay en 3 dimensiones, además de que las propiedades obtenidas en las mallas no son las mismas que en 2D [4].

Para poder abarcar lo anterior, se utilizará el motor de videojuegos Unity [5] para poder crear una esta herramienta interactiva. Un motor de videojuegos es un *framework* que facilita el desarrollo de videojuegos y aplicaciones, integrando librerías y otros programas al entorno de trabajo.

Las investigaciones en esta área han tenido distintos avances durante los últimos años, proponiendo, entre otras cosas, nuevos algoritmos para la mejora y manipulación de mallas geométricas. Con esto en consideración, se vuelve algo de interés el poder, en la misma herramienta, mostrar el resultado de este tipo de operaciones.

El anterior problema es algo que no ha pasado desapercibido en los trabajos de investigación de Rivara en esta área (por ejemplo, [2, 6–8]). Viendo la oportunidad de solventar este problema, se plantea la creación de una herramienta para poder visualizar mallas de tetraedros y que esta permita aplicar y observar operaciones en 3D que no tienen visualizadores hoy en día, suponiendo una ayuda para investigaciones y siendo una herramienta pedagógica para esta área.

# Capítulo 2

## Conceptos Previos

### 2.1. Mallas volumétricas

El objeto de interés en esta aplicación se centra en las mallas volumétricas, que son utilizadas ampliamente en computación gráfica. Estas mallas de 3 dimensiones son una representación tanto de la superficie del objeto como del interior de este, gracias a que están compuestas por cuerpos geométricos, encontrándose en su interior vértices, aristas y caras.

Existen varios tipos de mallas volumétricas utilizadas en la computación gráfica, cada una con sus correspondientes ventajas y desventajas según el contexto en donde se utilicen. Entre estas se encuentran las mallas prismáticas y hexaédricas, ambas utilizan prismas y hexaedros respectivamente para la representación volumétrica de los objetos. En la aplicación se utiliza la versión tetraédrica, en donde la malla está compuesta en su totalidad por tetraedros para la definición del volumen.

Las mallas volumétricas son útiles para la representación y el análisis de estructuras de 3 dimensiones. Cada elemento que conforma la malla posee un volumen y posición específica en el espacio, lo que permite realizar visualizaciones completas del objeto y hasta simular propiedades físicas de estos.

Estas mallas, al ser una forma versátil de representar datos de un modelo, permiten realizar distintos procesos a estas, ya sea utilizar algoritmos para analizarlas, procesarlas o modificarlas. Esto facilita realizar una variedad de cosas, como la simulación de fenómenos físicos o modelamiento de estructuras. En lo que concierne a este trabajo, se utilizan estas mallas para la creación de un visualizador.

### 2.2. Triangulación de Delaunay

La triangulación de Delaunay es un tipo de triangulación utilizado tanto en matemáticas como en la geometría computacional que cumple la condición de Delaunay. Esta condición indica que la circunferencia circunscrita de cada triángulo de la triangulación no debe contener ningún vértice de otro triángulo.

Llevando esto a su versión 3D, para una malla tetraédrica, la triangulación de Delaunay nos dice que todo tetraedro de la red no está contenido en la esfera circunscrita de ningún otro tetraedro.

**Definición 2.1** *Una triangulación de malla volumétrica cumple la condición de Delaunay, si para todo tetraedro  $t_0$  perteneciente a la malla, su esfera circunscrita no contiene vértices*



que no formen parte de  $t_0$ .

La triangulación de Delaunay tiene diversas propiedades que proporciona una forma eficiente de representar y analizar conjuntos de puntos, manteniendo la calidad y la coherencia de los elementos triangulares. Esto lo hace un objeto de gran interés en campos como la computación gráfica, la simulación física o la generación de mallas.

## 2.3. Arista terminal

En una triangulación, se dice que una arista es terminal cuando esta es la arista más larga de los triángulos que las comparten. En un espacio 2D, la arista terminal es la arista más larga que conecta los vértices de los triángulos. En el caso 3D de una malla volumétrica, es la arista más larga que conecta los vértices de los tetraedros que la comparten.

Esta arista posee gran valor de estudio en los usos de las mallas, ya sea para simulaciones de fenómenos físicos, como el modelado de objetos 3D.

En las mallas geométricas, la calidad es un aspecto de suma relevancia a la hora de ser utilizadas. La calidad de la malla, es un indicador de eficiencia de cuan bien se está representando la geometría del objeto en cuestión. Da a conocer la uniformidad y precisión del mallado, así una buena malla tiene guardados con detalles las partes importantes de la geometría, disminuyendo la imprecisión de la representación.

Dentro de esto, la longitud del borde terminal es un indicador de que tan buena es la calidad del mallado, ya que en la arista terminal no debe tener una longitud mucho más mayor que los demás bordes. La alta variabilidad de longitud en las aristas de una triangulación puede llevar a una malla no bien formada, con errores o distorsiones para sus usos.

Con lo anterior, podemos entender como los bordes terminales son de gran interés en el refinamiento y mejoramiento de mallas geométricas o volumétricas. Así, uno puede saber en qué lugar de la representación se necesita una mayor resolución y precisión de los datos, pudiendo llevar operaciones con el fin de aumentar la calidad de la malla y llevar a cabo su uso de mejor manera.

Existen otras utilidades que nos proveen las aristas terminales, como el entender como estas pueden afectar la precisión y estabilidad de las simulaciones en mallas o como estos bordes son utilizados en algoritmos de optimización para mallas.

## 2.4. Estrella terminal

Ya habiendo visto como las aristas terminales son de gran utilidad para la aplicación de este trabajo, tenemos que entender un concepto clave que surge en las mallas volumétricas.

La estrella terminal es el conjunto de cuerpos geométricos de la malla tal que cada uno de ellos comparte la arista terminal. En el caso de este trabajo son los tetraedros que comparten la arista terminal del mallado.

Entenderemos la existencia de 2 tipos de estrellas terminales [9]:

- Estrella cerrada: la arista terminal está en el interior de la malla.
- Estrella abierta: la arista terminal está en el exterior de la malla.

Al igual que en 2D la arista terminal son importantes, las estrellas terminales son también de gran importancia dentro del área. Si tomamos en cuenta trabajos como los hechos por Ri-

vara, Rodriguez y Balboa [2, 6, 8] en donde se estudian operaciones sobre mallas geométricas, vemos un interés en los algoritmos que actúan sobre la estrella terminal, buscando mejorar la calidad de las mallas.

Las estrellas terminales también tienen un papel importante en la generación de mallas 3D y en las simulaciones numéricas por computadora.

De la misma manera que antes, las estrellas terminales toman un gran papel en el análisis de calidad de estas mallas, este conjunto de, en este caso, tetraedros facilitan una manera de evaluar la calidad de la malla. Homólogamente, si la estrella terminal posee una arista terminal con longitud mucho mayor a las otras aristas del mallado, puede ser que la triangulación no haya sido de buena calidad.

Siguiendo con esto, como la estrella terminal nos proporciona una gran cantidad de información sobre la calidad de la malla, podemos utilizarla para refinar de una buena y eficiente manera el mallado, reduciendo así la variabilidad de los largos de las aristas y logrando una mayor precisión de la malla en lugares donde es requerido.

Además, estos conjuntos son tomados en cuenta para otras cosas, como en los algoritmos de optimizaciones de mallas y en los análisis de precisión en la simulación, al igual que su estabilidad.

Con todo lo anterior podemos entender el contexto del trabajo, viendo así los objetos a utilizar, como la malla, sus componentes y los subconjuntos de esta que son de interés (arista y estrella terminal).

## 2.5. Algoritmos LEPP

Ahora, se tiene claro que tetraedros del conjunto son de mayor interés, pero ¿cómo podemos identificar dichos cuerpos geométricos en la malla? Para esto existen algoritmos que permiten encontrar la estrella terminal de un conjunto.

Los algoritmos LEPP (por sus siglas en inglés *Longest Edge Propagation Path*) son algoritmos que operan sobre mallas geométricas o volumétricas que dado, en este caso, un tetraedro inicial va avanzando sobre la malla a través de los vecinos del tetraedro por la arista más larga de estos.

Notar que en 3D, las aristas pueden ser compartidas por múltiples tetraedros, lo que conlleva que LEPP se propague por distintos caminos, pudiendo así encontrar un conjunto de aristas terminales y estrellas terminales.

El funcionamiento de este algoritmo es tal que el LEPP de un tetraedro  $t_0$  recorre la triangulación de la malla partiendo por  $t_0$ , tomando su arista más larga y preguntando en todos los tetraedros que la comparten cuál es la arista más larga. Este proceso se repite exhaustivamente, encontrando así un conjunto de aristas terminales de la malla y por consiguiente, las estrellas terminales de estas [6, 8].

Si bien existen distintas versiones de este algoritmo, la que nos compete para este trabajo es una versión simple del algoritmo LEPP [9]. En esta versión se parte por buscar la arista de mayor tamaño del tetraedro a utilizar, luego se revisan los tetraedros que comparten esta arista para luego tomar solo el tetraedro que posea la arista más larga (en caso de existir empate, se elige uno al azar) y así continuar con este tetraedro la búsqueda de la arista terminal. Esto finalizará cuando la arista más larga de los vecinos sea la misma, la cual todos estos comparten.

Esto simplifica en gran medida el nivel de complejidad del algoritmo, como también el nivel de cálculos que se deben realizar, ya que en su versión completa, se debe avanzar por todos los

tetraedros que posean una arista mayor, de esta manera el cálculo de este algoritmo puede complicarse rápidamente dependiendo de la malla y tetraedro en cuestión. Cabe destacar que en esta versión del algoritmo, la arista terminal encontrada podría no ser la arista más larga de toda la triangulación, encontrado así solo un máximo local y no global, sin embargo, esto no es importante para este trabajo, ya que la visualización tiene el fin de ilustrar el funcionamiento de operaciones de mallas.

## 2.6. Operaciones

En las mallas geométricas y volumétricas existen una variedad de operaciones que se pueden llevar a cabo para manipular y analizar datos en entornos bidimensionales o tridimensionales. Estas operaciones son esenciales para diversas aplicaciones, desde la simulación numérica hasta la animación por computadora.

Las siguientes operaciones son claves para el refinamiento y mejoramiento de mallas

### 2.6.1. Inserción de punto

La inserción agrega un nuevo vértice en la malla, transformando un conjunto de tetraedros.

### 2.6.2. Suavizamiento

El *smoothing* o suavizado es una de las operaciones a implementar y se trata del traslado de un punto del mallado para así lograr una mejor triangulación.

### 2.6.3. Intercambio de aristas

Esta última operación consiste en una reestructuración de un conjunto de tetraedros con una arista en común, cambiando este conjunto por otro, en donde se elimine dicha arista y sea sustituida por otras nuevas.

## 2.7. Motor de videojuego

Los motores de videojuegos sirven como herramientas de gran ayuda para el desarrollo de videojuegos y aplicaciones interactivas. Gracias a estos, se pueden lograr con mayor sencillez distintas tareas del proceso de creación, como, por ejemplo, la renderización de los gráficos, facilitando el trabajo con texturas y modelos. Además, incorporan una gran cantidad de técnicas para los efectos especiales, iluminación y sombras, con lo que se logran gráficos de gran nivel.

Dentro de la renderización, el motor trabaja con distintos tipos de elementos, como los modelos 3D, los cuales describen un conjunto de polígonos que conforman la representación del objeto a modelar. También existen las texturas, que son imágenes que se aplican a las superficies de los modelos. Otro de estos elementos son los *shaders*, que son programas que manejan como pintar modelos o escenas, generando así efectos visuales. Por último, mencionar que dentro de esto, el motor también se encarga de la cámara, objeto el cual se encarga de mostrar lo que el usuario mire dentro de la aplicación.

Existe también un nivel de abstracción que consigue el desarrollador con relación al hardware de la máquina en donde se ejecutará la aplicación. Es el motor quien se encarga de comunicarse con los componentes de esta, permitiendo así simplificar inmensamente el pro-

ceso de desarrollo y compilación para distintas plataformas.

Los motores de videojuegos también manejan los recursos utilizados dentro del programa, gestionando así las cargas y descargas de recursos a memoria del sistema para poder obtener rendimientos deseables en una aplicación de formato interactivo. Dentro del manejo de recursos también existen optimizaciones que permiten aprovechar estos de mejor manera.

También existen funcionalidades incluidas en estos programas para el manejo de entrada, pudiendo así facilitar la generación de buenas interfaces gráficas para que los usuarios sean capaz de interactuar con el programa.

El trabajo utiliza el motor de videojuegos Unity [5]. Este software es de los más conocidos y utilizados dentro del rubro, teniendo así una gran comunidad de desarrolladores. Además, no requiere de pagos para poder empezar a programar con todas sus funcionalidades. Unity es una herramienta que se especializa tanto en aplicaciones 2D como en 3D.

Unity se caracteriza por ser una aplicación de fácil uso que permite la creación de software para PC o consolas de videojuegos. Esta herramienta numerosas funcionalidades que hacen que Unity se haya catalogado como una excelente herramienta para el desarrollo de videojuegos.

El motor permite el *scripting* en el lenguaje de programación C#. Con esto es posible crear la lógica de la aplicación, manejar los eventos que se efectúan dentro del videojuego y permitir el uso de distintas librerías y APIs del motor.

# Capítulo 3

## Fundamentación del tema

### 3.1. Avances del área

El campo de las mallas geométricas y volumétricas es utilizado en diferentes áreas de la ciencia, la ingeniería y la computación. A lo largo del tiempo, el estudio en el área ha logrado avances, creando soluciones eficientes a problemas complejos.

Para lo anterior, se han desarrollado e investigado técnicas y algoritmos para optimizar y mejorar la calidad de las mallas, obteniendo así buenos mallados capaces de representar distintos objetos.

La generación y visualización de mallas volumétricas son tareas con una gran cantidad de cálculo computacional, por lo que es normal ver que existan grandes esfuerzos por desarrollar algoritmos eficientes para estas tareas [6].

En relación con lo anterior, las mallas han objetos de distintas operaciones. Sin embargo, estas operaciones son complejas y entenderlas es de gran importancia para el estudio del área.

Algunas de estas operaciones permiten un refinamiento de la malla, modificando la triangulación, ya sea ajustando la posición de los puntos, cambiando las aristas que componen el mallado o agregando más vértices al conjunto.

Comprender cómo funcionan estas operaciones de maneras más simples y eficientes es importante para el estudio del área, ya que impulsa el interés en el área y fomenta futuras investigaciones.

### 3.2. Visualizadores de mallas

El problema de visualizar y manipular mallas volumétricas no es algo sencillo de resolver, pero tampoco es un desafío sin precedentes. Existen diversos programas que trabajan con este tipo de elementos, como los programas CAD [10] o las aplicaciones para modelar gráficos en 3 dimensiones.

Además, el estado del arte de los visualizadores de mallas ha experimentado un gran desarrollo gracias a los avances en gráficos de computadora y tecnologías de procesamiento paralelo. Los visualizadores de mallas actuales ofrecen una amplia gama de características y capacidades que permiten una representación detallada y precisa de grandes datos geométricos.

Estas herramientas permiten una visualización interactiva, lo que brinda una experiencia de usuario altamente inmersiva al permitir la rotación, traslación y el desplazamiento fluido de los objetos, facilitando la exploración detallada de la geometría. También poseen

técnicas avanzadas de renderización para producir imágenes realistas y de alta calidad, como sombreado suave, efectos de iluminación global y sombras dinámicas.

Gracias a la gran capacidad de procesamiento y memoria de los computadores actuales, los visualizadores actuales pueden manejar grandes conjuntos de datos sin comprometer el rendimiento. Además, algunos visualizadores se han adaptado para funcionar con tecnologías emergentes como la realidad virtual y aumentada, brindando a los usuarios una experiencia inmersiva al interactuar con modelos tridimensionales.

Sin embargo, los visualizadores de mallas actuales tienen limitaciones cuando se trata de estudiar en detalle las operaciones de mallas y cómo funcionan internamente. Si bien pueden proporcionar una representación visual útil de los resultados de las operaciones, es necesario combinarlos con otras herramientas y recursos para una comprensión más profunda de éstos.

Siguiendo con lo anterior, las herramientas mencionadas, en su mayoría de carácter comercial, presentan un enfoque distinto al que se busca en este trabajo. Estas herramientas no tienen como objetivo principal el estudio de mallas, son complejas y de difícil uso.

Entre estas herramientas, se encuentra MeshLab [11], que permite la manipulación de mallas geométricas, pero no se enfoca en ilustrar operaciones de estrellas terminales, es más automatizada y oculta el detalle de las operaciones realizadas. También están los motores de videojuegos, como Unreal Engine, que utilizan las mallas geométricas para otros fines, principalmente enfocándose en la visualización de estas.

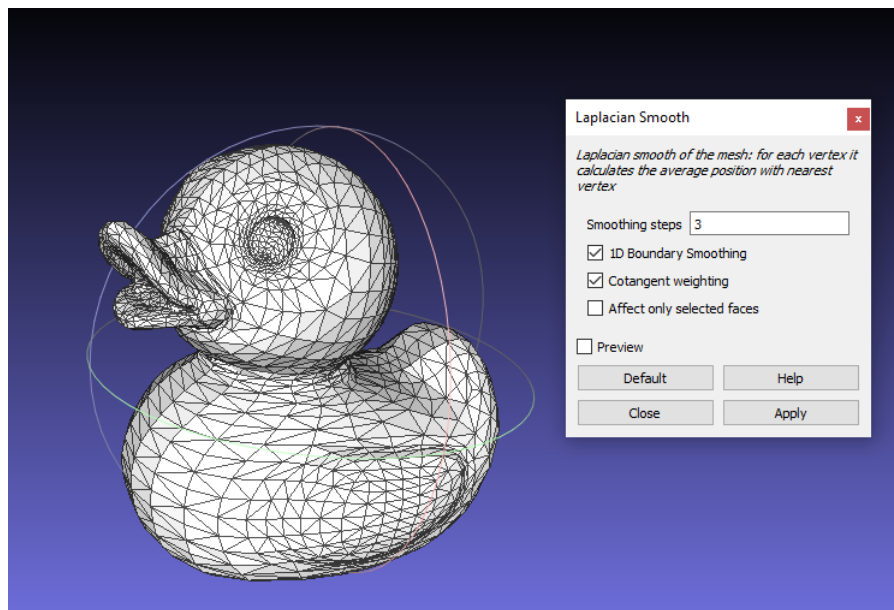


Figura 3.1: MeshLab.

Como se mencionó antes, las herramientas existentes tienen propósitos más amplios y son programas de alto nivel, por lo que no se enfocan en operaciones más atómicas como inserciones simples o suavizamiento de puntos. Como resultado, los efectos directos de estas operaciones no se aprecian del todo.

### **3.3. Herramientas didácticas**

Existe un déficit de visualizadores que faciliten la enseñanza de las operaciones en mallas tetraédricas y, al mismo tiempo, que sean de gran utilidad para los investigadores. La visualización tridimensional puede ser de gran ayuda en la comprensión de conceptos complejos y en la capacidad de manipular y analizar datos en entornos 3D.

Los visualizadores de mallas son herramientas visuales poderosas que permite explicar conceptos abstractos de una manera más accesible y práctica. Al mostrar visualmente las operaciones en mallas geométricas y volumétricas, es posible comprender mejor cómo funcionan estas operaciones y cómo afectan la estructura de las mallas, permitiendo un entendimiento más intuitivo de conceptos abstractos y facilitando la conexión entre la teoría y la práctica.

La retroalimentación en tiempo real que brindan estos visualizadores durante la ejecución de operaciones en la malla es una característica beneficiosa para los estudiantes, ya que les permite ver instantáneamente los resultados y comprender cómo cada cambio afecta la representación tridimensional. Esta capacidad acelera el proceso de aprendizaje y ayuda a detectar posibles errores.

Con esta rápida experimentación y la interactividad en el visualizador, se permite una experiencia de aprendizaje más inmersiva y enriquecedora al tener la capacidad de explorar los modelos tridimensionales rápidamente. Esto resulta en una enseñanza más efectiva y estimulante en el campo de las mallas geométricas y volumétricas.

Los visualizadores de mallas pueden ser una herramienta valiosa para analizar y comprender los resultados de estudios. También pueden ser utilizados para ilustrar mallas de estudios y comportamientos de estas al aplicarle distintas operaciones.

# Capítulo 4

## Diseño de la aplicación

### 4.1. Estructura de datos

La aplicación requiere de un método sencillo, eficaz y organizado para poder obtener acceso a los datos procedentes de la malla geométrica y, así, realizar un buen renderizado de esta y poder ejecutar las operaciones correctamente y de manera eficiente. Para solventar esto, se utiliza una estructura de datos simplificada basada en el trabajo de Pedro Rodríguez [1].

En dicho trabajo, se tiene en cuenta que la malla posee 4 elementos: vértices, aristas, caras y tetraedros. Los tetraedros, a su vez, se componen por 4 vértices ( $v_0, v_1, v_2, v_3$ ), 6 aristas ( $e_0, e_1, e_2, e_3, e_4, e_5$ ) y 4 caras ( $f_0, f_1, f_2, f_3$ ) y, tal como se muestra en la Figura 4.1, el vértice  $v_i$  está en el lado opuesto de la cara  $f_i$ . En la malla, los tetraedros solo pueden compartir a lo más un vértice, una arista o una cara y estos no pueden superponerse entre ellos.

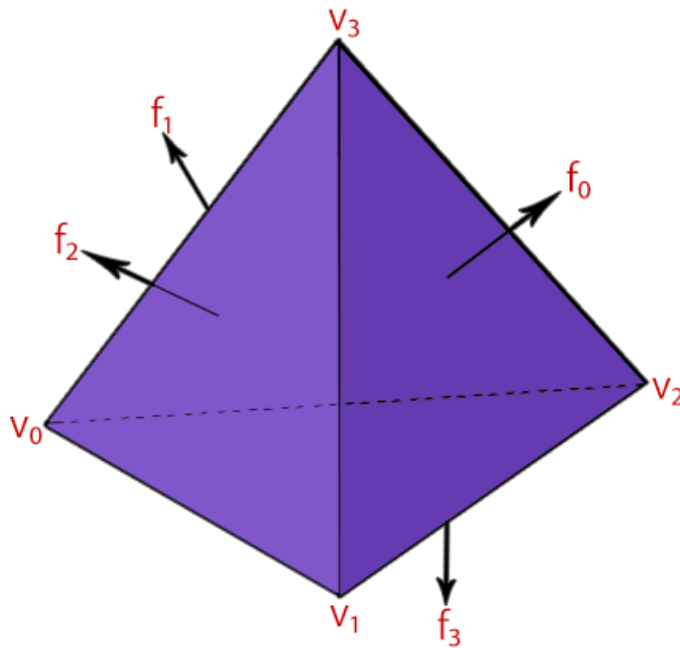


Figura 4.1: Representación de tetraedro.



Con la anterior representación, los tetraedros deberán tener siempre una orientación, en este caso, positiva. Esto es, dado un tetraedro  $t_i$  con orientación positiva, se tiene que  $\forall v_i \in t_i$ ,  $v_i$  está en el lado positivo de los demás vértices (recorriéndolos en sentido antihorario).

Para este trabajo se usará una versión simplificada de lo anterior. Se tendrá como estructura de datos una en donde se guarde la malla como un conjunto de tetraedros. En cada uno de estos se guardarán:

- Los vértices: estos elementos de la malla están representados por puntos en el espacio 3D. Cada vértice almacena sus coordenadas (x, y, z). Con esto se tendrá la información necesaria para dibujar el tetraedro.
- Las aristas: compuestas por los vértices que lo componen y la longitud de esta. De gran utilidad para facilitar la utilización del algoritmo LEPP.
- Los vecinos: Arreglo de los tetraedros que están unidos por medio de una cara (3 vértices).

Esta estructura de datos permite acceder fácilmente a la información de cada tetraedro, sus vértices y sus tetraedros vecinos. Al tener referencias a los vértices y tetraedros vecinos en cada objeto de tetraedro, se puede navegar eficientemente a través de la malla.

Además, esta estructura de datos no solo permite generar una visualización de la malla en la aplicación, sino que posee las suficientes características para poder realizar de buena manera las operaciones deseadas sobre esta de una manera eficiente al acceder rápidamente a la información necesitada.

Esta estructura de datos es escalable y mantenible a medida que se agregan o eliminan tetraedros en la malla. Además, debido a su naturaleza orientada a objetos del programa, permite una mayor flexibilidad en la adición de atributos o métodos adicionales que pueden ser útiles para necesidades específicas del proyecto.

Si bien, la elección de una estructura de datos adecuada depende de las características y requerimientos de la aplicación, la naturaleza de este trabajo con mallas 3D hacen de esta estructura una elección sólida y bien adaptada a las necesidades de la aplicación.

## 4.2. Manejo de archivos

El manejo de archivos de esta herramienta visualizadora se limita solo a archivos con extensión .m3d con el fin de no complicar en exceso el tamaño de este proyecto y focalizar esfuerzos en la visualización y modificación del modelo como tal.

Se escoge esta extensión, ya que es ampliamente conocida dentro de los tipos de archivos para representar modelos 3D, siendo utilizado ampliamente en la impresión, pero con suficiente información para poder utilizarse en otros contextos.

Este formato es un texto que contiene en líneas las distintas características del modelo que representa, cada línea posee un identificador en su primer carácter que permite reconocer que información es la que se está entregando.

Posee la información de los vértices del modelo, almacenando su posición en el espacio de 3 dimensiones, y un identificador que sirve posteriormente para la descripción de los tetraedros.

También posee una lista de los tetraedros que conforman la malla en sí, describiéndolos como 4 identificadores correspondientes a los vértices que lo conforman y un identificador del tetraedro como tal.

Finalmente, se describen los vecinos que poseen los tetraedros en cada línea, nombrándolos por su identificador y señalando sus vecinos con esto mismo.

Estos elementos son leídos línea a línea mediante un Script y almacenados en distintos arreglos para cada una de estas características. Luego, son utilizados para la creación de los tetraedros, según como se describió anteriormente en la estructura de datos. Los tetraedros finalmente son guardados en un arreglo que representará a la malla volumétrica.

### 4.3. Visualización de tetraedros

La visualización de tetraedros implica representar estos elementos geométricos de una manera clara y comprensible en un entorno tridimensional.

Los tetraedros son visualizados como formas sólidas en el espacio 3D. Cada tetraedro se muestra mediante sus caras y aristas, lo que permite a los usuarios comprender la forma y estructura del objeto en estudio.

Se utilizan colores para resaltar características específicas o atributos de la malla. Esto será particularmente útil en las operaciones, ya que afectan diferentes tetraedros y es crítico poder identificarlos de una manera sencilla.

El usuario con la visualización podrá navegar en tiempo real en la malla de tetraedros. Los usuarios podrán rotar, mover y acercar la visualización para obtener diferentes perspectivas y explorar la distribución espacial de los tetraedros.

Una función sobresaliente de esta herramienta es la visualización de los tetraedros internos al ajustar su tamaño. Los usuarios tienen la capacidad de modificar el tamaño de la representación, lo que permite la observación de la distribución interna de los tetraedros y permite descubrir detalles ocultos dentro del volumen.

### 4.4. Algoritmo LEPP

Se optó por implementar las 2 versiones descritas anteriormente del algoritmo LEPP.

#### 4.4.1. LEPP simple

En esta versión del algoritmo *Longest Edge Propagation Path* se simplifica la cantidad de caminos que pueden surgir, tomando solo el camino de la arista más larga de cada paso del algoritmo. Para esto se requiere un tetraedro de la malla previamente seleccionado desde el cual se dará inicio a este algoritmo.

El algoritmo parte buscando la estrella terminal en la vecindad de la arista más larga del tetraedro seleccionado, con esto se comprobará si esta arista es la más larga de todo el subconjunto, de ser así el algoritmo terminará habiendo encontrado la estrella terminal, de lo contrario, repetirá el proceso ahora con la vecindad de la nueva arista más larga.

---

**Algoritmo LEPP simple**

---

Sea  $t$  un tetraedro seleccionado de la malla.  
Definir  $l$  como la arista más larga de  $t$ .  
**while** true **do**  
    Definir  $E$  como el conjunto de vecinos de  $t$  por  $l$ .  
    Agregar  $t$  al LEPP.  
    **if** aristaMásLarga( $E$ ) >  $l$  **then**  
        Redefinir  $t$  como cualquiera de los tetraedros que posea la aristaMásLarga( $E$ ).  
    **else**  
         $E$  es la estrella terminal.  
        **Break.**  
    **end if**  
**end while**

---

Esta versión, al encontrar solo una estrella terminal, es la utilizada por las operaciones implementadas, simplificando el flujo de uso del usuario.

#### 4.4.2. LEPP completo

La siguiente versión es la implementación completa del algoritmo LEPP, recorriendo todos los múltiples caminos que puedan surgir.

Al igual que la anterior versión, el algoritmo parte del tetraedro seleccionado, propagándose por sus tetraedros vecinos a través de la arista más larga. Esta vez, se recorren todos los caminos, entregando así un conjunto de estrellas terminales.

---

**Algoritmo LEPP**

---

Sea  $T$  un conjunto de tetraedros, inicialmente un tetraedro seleccionado.  
**while** true **do**  
    Definir  $T^*$  como un conjunto vacío de tetraedros.  
    **for all**  $t \in T$  **do**  
        Agregar  $t$  al LEPP.  
        Definir  $E$  como el conjunto de todos los vecinos de  $t$  por la arista más larga.  
        **if** No hay tetraedro en  $E$  con una arista más larga **then**  
             $E$  es una estrella terminal.  
            **Break.**  
        **else**  
            Agregar los tetraedros con aristas más largas a  $T^*$ .  
        **end if**  
    **end for**  
    **if**  $T^*$  vacío **then**  
        **Break.**  
    **else**  
        Definir  $T$  como  $T^*$ .  
    **end if**  
**end while**

---

Mientras que la versión simple del algoritmo sirve para entender el uso de este algoritmo

con otras operaciones, la versión completa ayuda a comprender el funcionamiento de un algoritmo LEPP en mallas 3D.

## 4.5. Operaciones

La herramienta interactiva creada debe poder realizar sobre la malla las siguientes operaciones LEPP de interés.

### 4.5.1. Inserción de punto

Tal como se explica en [6] en la inserción de centroides, primero se parte con un algoritmo LEPP para encontrar la estrella terminal, para luego calcular el centroide de esta y así poder realizar la inserción del nuevo punto en la malla.

En esta implementación se toma la estrella terminal resultante del algoritmo LEPP con la cual se itera sobre cada tetraedro  $t_0$  perteneciente a la estrella, reemplazando  $t_0$  por 2 nuevos tetraedros resultantes de unir las 2 caras que no contienen la arista terminal con el nuevo punto. Lo que consigue un reemplazo de la estrella terminal por otro conjunto de tetraedros en donde este nuevo punto forme parte.

El punto a insertar es el centroide de la estrella terminal, exceptuando si la estrella terminal es abierta, en tal caso se utiliza el punto medio de la arista terminal.

---

#### Algoritmo Inserción de centroide

---

Sea  $\mathbf{S}$  una estrella terminal

**if**  $\mathbf{S}$  es una estrella abierta **then**

    Definir  $p$  como el punto medio de la arista terminal de  $\mathbf{S}$ .

**else**

    Definir  $p$  como el centroide de  $\mathbf{S}$ .

**end if**

**for all**  $t \in \mathbf{S}$  **do**

    Remover la arista terminal de  $t$ .

    Sean  $f_1$  y  $f_2$  las caras restantes de  $t$ , Crear los tetraedros  $t^*$  y  $t^{**}$  con  $f_1$  y  $f_2$  respectivamente, uniéndolos con  $p$ .

    Agregar a  $t^*$  y  $t^{**}$  a la malla.

    Remover a  $t$  de la malla.

**end for**

---

### 4.5.2. Suavizamiento

La operación de suavizamiento (*smoothing* en inglés), tiene como entrada un vértice al cual se le realizará una transformación a su posición. Dicha transformación viene dada por el centroide del conjunto a transformar, por lo que se calcula el centroide de los demás vértices. Con esto se cambiará un conjunto de tetraedros por otro, en donde el vértice de interés ha sido cambiado por el centroide calculado.

---

**Algoritmo** Suavizamiento

---

Sea  $\mathbf{S}$  un conjunto de tetraedros con un vértice  $p$  en común y  $\mathbf{V}$  su conjunto de vértices.  
Definir  $c$  como el centroide de  $\mathbf{V}$   
**for all**  $t \in \mathbf{S}$  **do**  
    Sea  $v$  los vértices de  $t$ , definir  $v^*$  como  $(v - p) \cup c$   
    Crear un nuevo tetraedro  $t^*$  cuyos vértices sean  $v^*$   
    Agregar  $t^*$  a la malla.  
    Remover  $t$  de la malla.  
**end for**

---

### 4.5.3. Intercambio de aristas

Otro tipo de operación de estudio en este campo es el intercambio de aristas [12, 13], algunos trabajos también buscan mejorar la triangulación al realizar intercambio de aristas en la estrella terminal [6]. Al igual que antes, se calcula la estrella terminal de un subconjunto de tetraedros con un algoritmo LEPP. En una estrella terminal se pueden realizar distintos intercambios de aristas, cambiando el subconjunto de tetraedros.

En función del alcance del trabajo realizado, se optó por que la herramienta soporte intercambios de aristas en estrellas terminales de 2 tetraedros y de 4.

---

**Algoritmo** Intercambio de arista

---

Sea  $\mathbf{S}$  una estrella terminal,  $\mathbf{V}$  sus vértices y  $l$  su arista terminal  
**if**  $\mathbf{S}$  tiene solo 2 tetraedros **then**  
    Definir  $a$  y  $b$  como los vértices de  $l$ .  
    Crear tetraedro con los vértices  $\mathbf{V} - a$ .  
    Crear tetraedro con los vértices  $\mathbf{V} - b$ .  
    Remover los tetraedros originales.  
**else if**  $\mathbf{S}$  tiene solo 4 tetraedros **then**  
    Separar los tetraedros en 2 grupos de 2 tetraedros que compartan 1 cara.  
    Realizar intercambio de arista con cada grupo.  
**end if**

---

## 4.6. Interfaz Gráfica

La interfaz gráfica ofrece una experiencia interactiva y comprensible para los usuarios, permitiéndoles explorar y analizar la malla de tetraedros de manera rápida y sencilla. A través de la interfaz, los usuarios pueden seleccionar un tetraedro específico de la malla y acceder a diversas opciones y herramientas para llevar a cabo distintas operaciones.

La primera funcionalidad es permitir al usuario seleccionar un tetraedro de la malla. Esto se puede lograr haciendo click en el tetraedro deseado en la visualización 3D.

Una vez que se ha seleccionado un tetraedro, el usuario tiene la opción de ejecutar el algoritmo LEPP. Con esto, la aplicación realizará el cálculo del par de puntos más cercano dentro del tetraedro seleccionado, lo que puede ser útil en aplicaciones de análisis estructurales o procesamiento de datos en geometría computacional.

El usuario tiene la capacidad de cambiar el tamaño de los tetraedros en la visualización,

lo que facilita la visualización de detalles internos de la malla.

La interfaz permite al usuario acceder a una serie de operaciones disponibles, como la inserción de centroide o mostrar los vecinos de un conjunto de tetraedros.

Para facilitar el uso de la aplicación y brindar retroalimentación al usuario, la interfaz incluye una ventanita que muestra distintos mensajes y notificaciones. Estos mensajes pueden ser informativos, como confirmaciones de operaciones realizadas o advertencias en caso de errores. De esta manera, el usuario siempre estará al tanto del estado de la aplicación y de las acciones que ha llevado a cabo.

Con estas funcionalidades, se espera poder transmitir una experiencia completa, intuitiva y sencilla para la visualización de los tetraedros y las operaciones que dispone la herramienta.

# Capítulo 5

## Resultados

En esta parte se muestran distintas vistas de la aplicación en donde se ilustran las distintas funcionalidades de esta

La aplicación cuenta con un menú inicial en donde el usuario puede elegir entre cerrar el programa o ir a la selección de mallas disponibles.

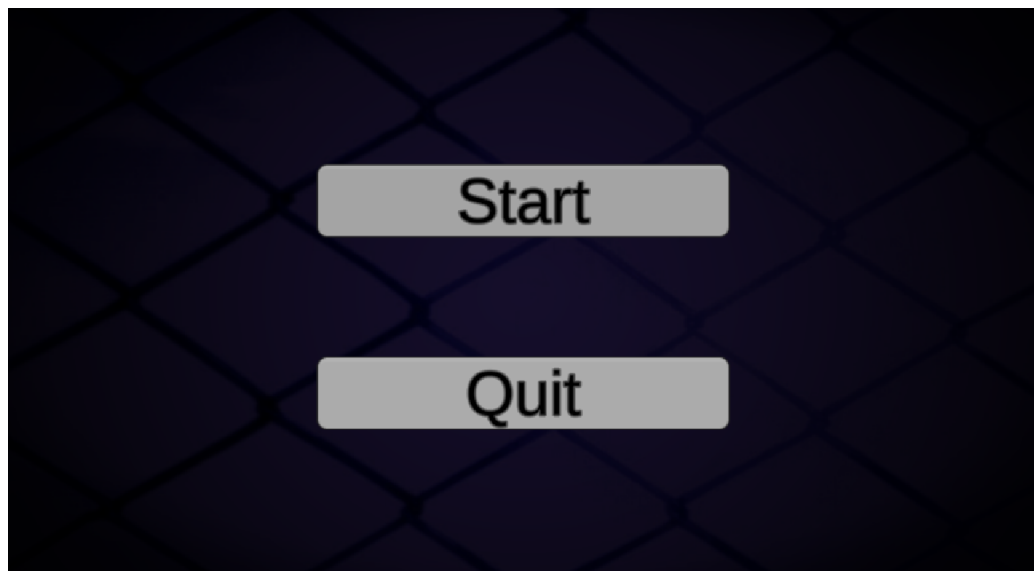


Figura 5.1: Pantalla inicial.

Seguido de la anterior pantalla, el usuario entra al menú de selección de mallas disponibles. Las mallas fueron provistas por Fernando Balboa [9]. En esta sección se puede elegir si volver a la vista anterior o elegir cuál modelo visualizar.

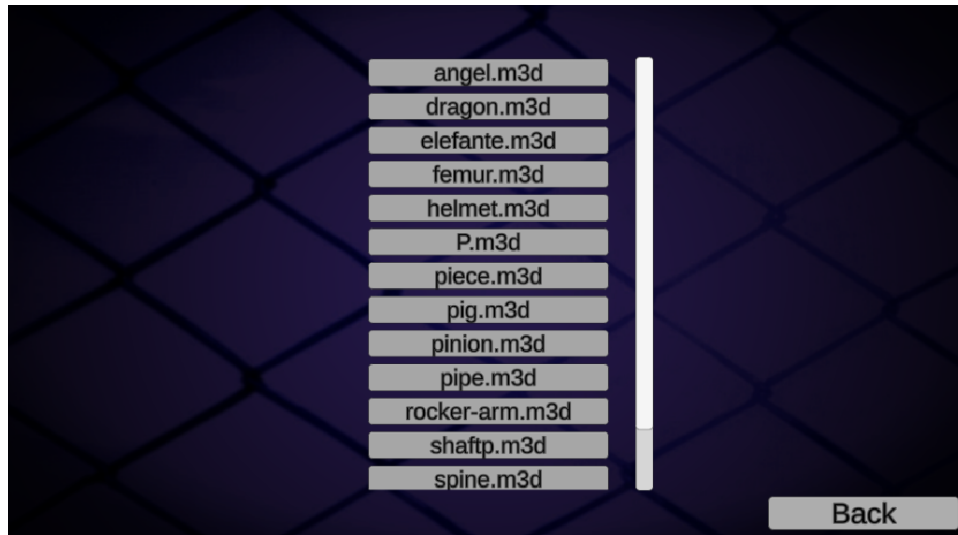


Figura 5.2: Menú de selección.

La siguiente pantalla es en donde se realiza la visualización como tal. Aquí el modelo seleccionado en la pantalla anterior es renderizado en pantalla.

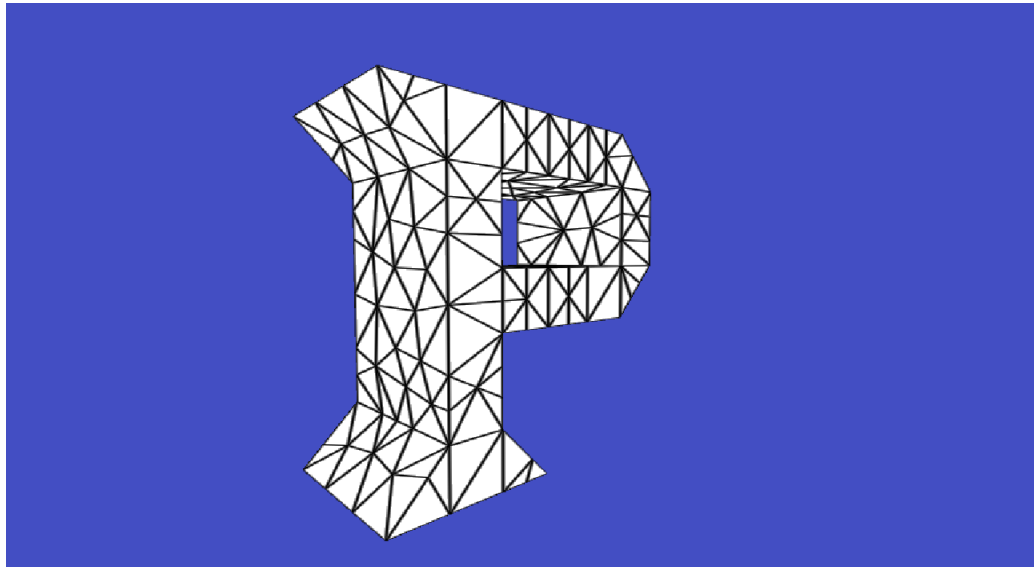


Figura 5.3: Visualización.



Existe un menú de opciones y una ventana de mensajes en la misma pantalla, para que así tener al alcance las distintas funcionalidades de la herramienta interactiva. El usuario es capaz de navegar por el modelado a través del mouse y el teclado.

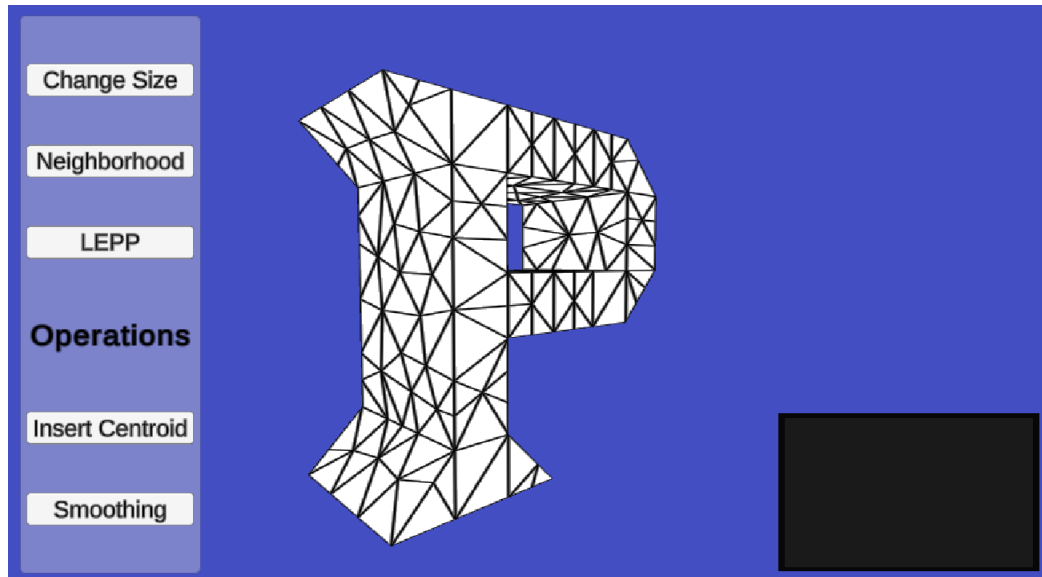


Figura 5.4: Visualización con menú.

Cabe mencionar que existe un menú de pausa al presionar la tecla ESC, la cual nos lleva a un menú en donde se puede elegir volver al menú de selección, mallas, salir del programa o volver a la visualización.

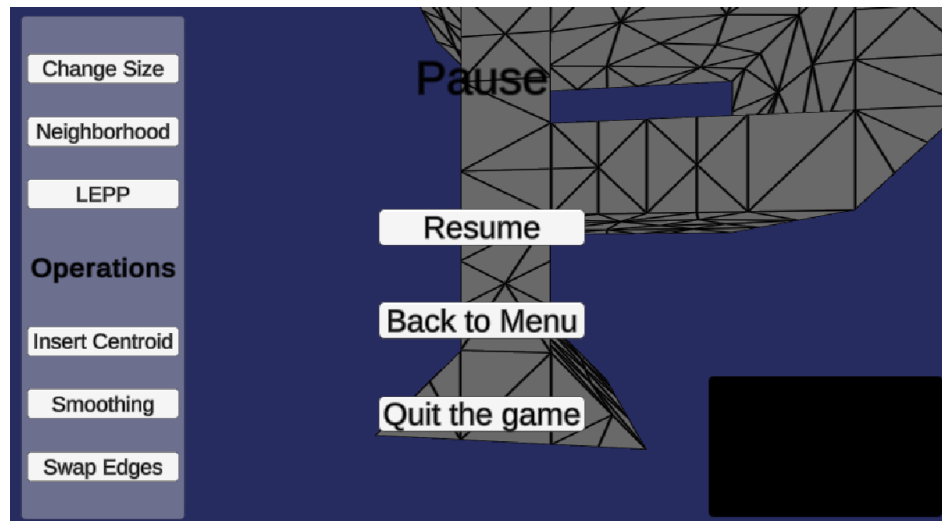


Figura 5.5: Menú de pausa.

Ahora, pasando a las funcionalidades de la aplicación, vemos que tenemos la opción de poder cambiar el tamaño a los tetraedros, logrando así poder visualizar la estructura interna de la malla volumétrica.

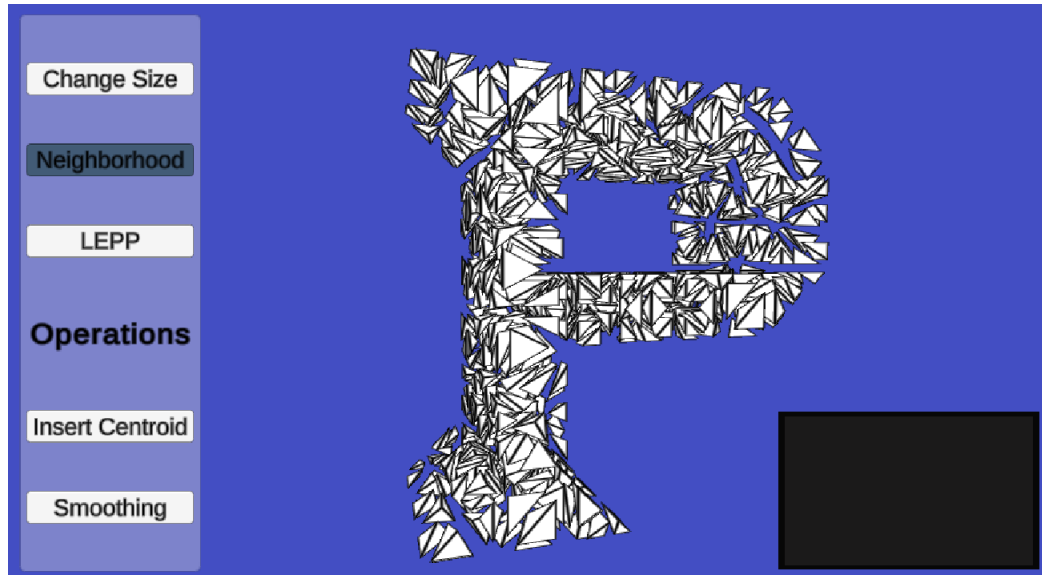


Figura 5.6: Visualización con tamaño modificado.

Como se tenía visto, la herramienta permite seleccionar los tetraedros mediante el click sobre ellos, marcándose de un color distinto.

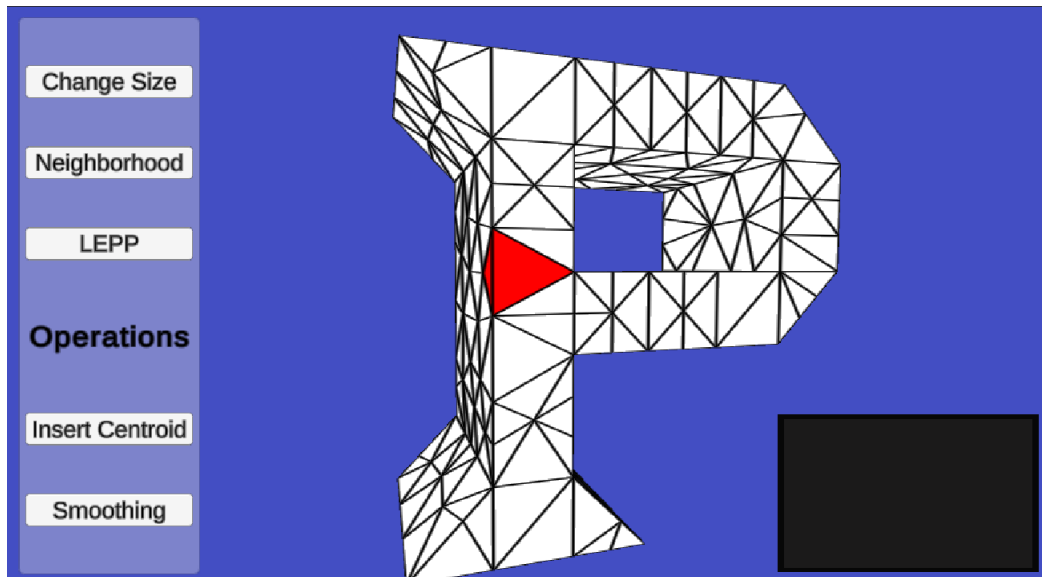


Figura 5.7: Selección de tetraedro.

Al tener un tetraedro seleccionado, es posible realizar otras 3 opciones:

- Revisar la vecindad de este, marcando con otro color los vecinos de este

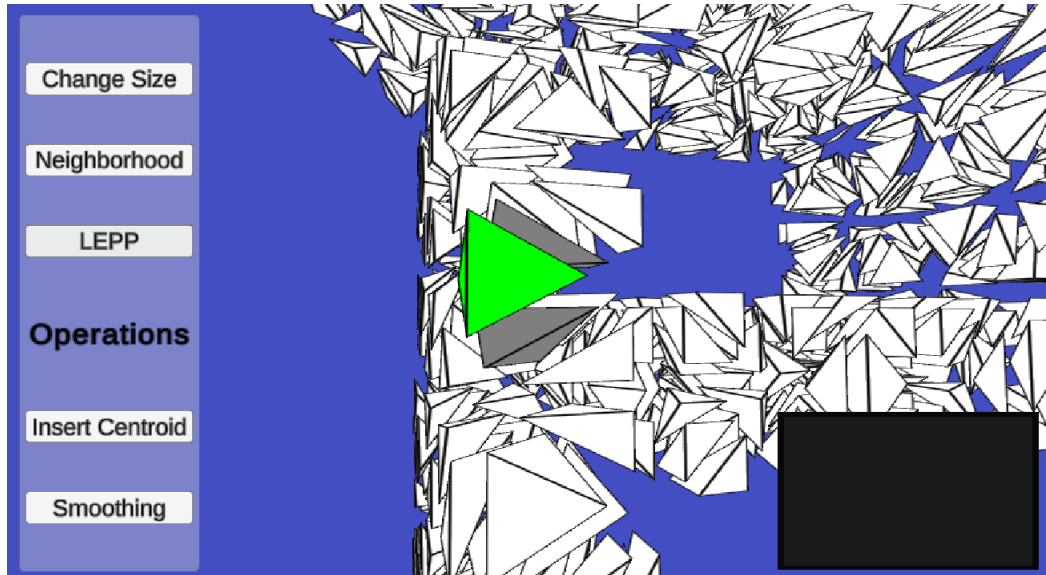


Figura 5.8: Vecinos de un tetraedro.

- Ejecutar LEPP sobre este tetraedro, marcando cada paso del cálculo con distintos colores y finalizando con un destaque a la arista terminal de la estrella.

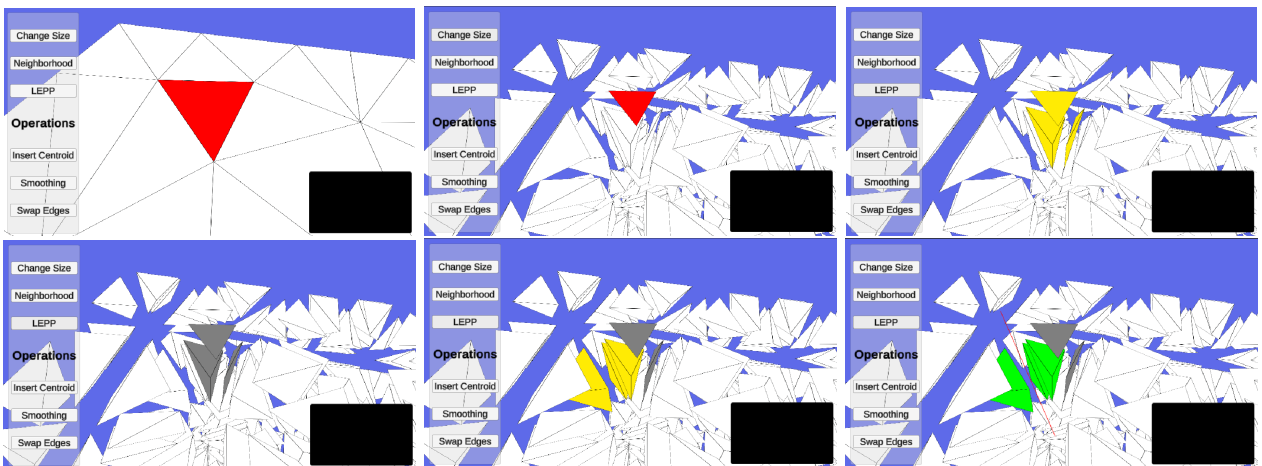


Figura 5.9: Algoritmo LEPP simple.

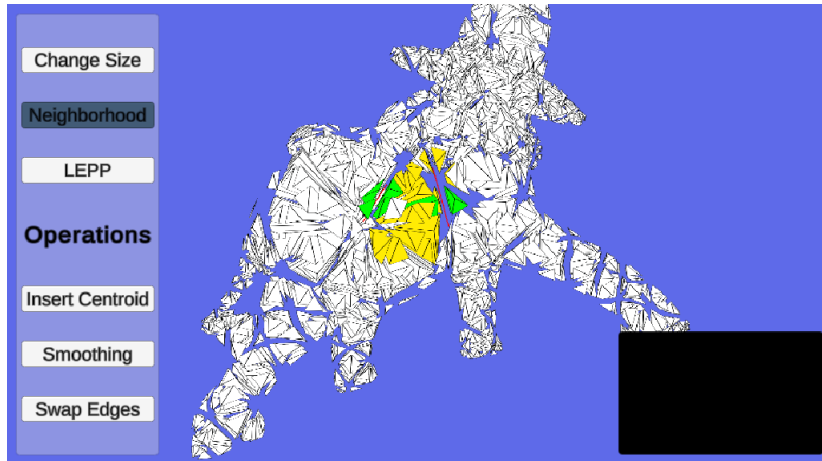


Figura 5.10: Algoritmo LEPP completo.

El usuario mediante el menú de opciones es capaz de realizar las operaciones sobre la malla, cambiando el conjunto de tetraedros y modificando la visualización de la malla. Todo esto destacándose con colores.

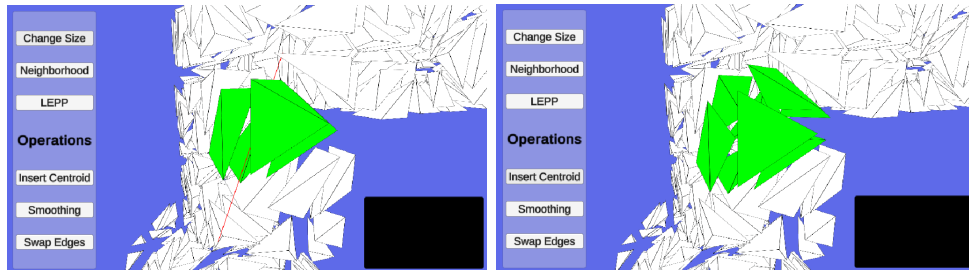


Figura 5.11: Inserción de centroide.

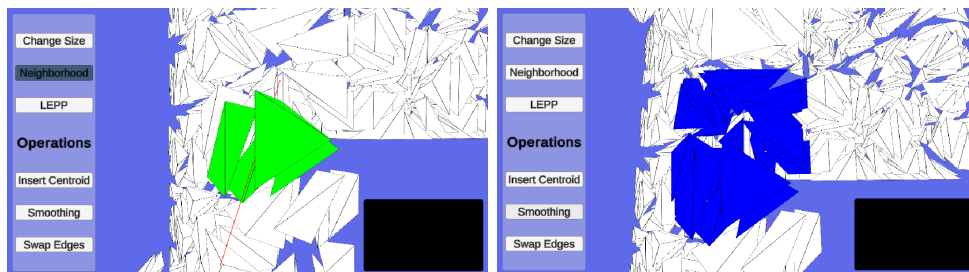


Figura 5.12: Suavizamiento.

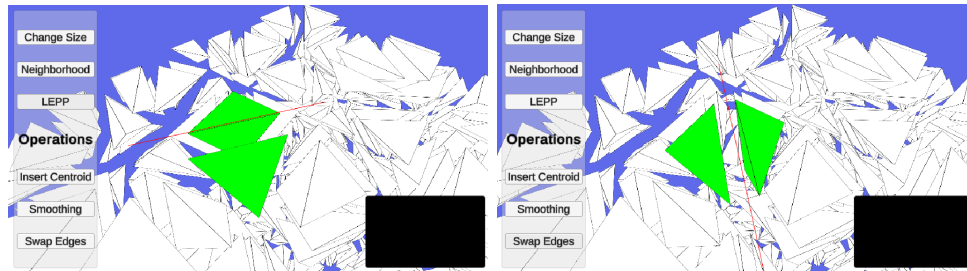


Figura 5.13: Intercambio de arista.

Es posible obtener los vecinos de un conjunto de tetraedros que han sido seleccionados, luego de una operación, obteniendo así todos los vecinos de los tetraedros seleccionados, sin contar a estos mismos.

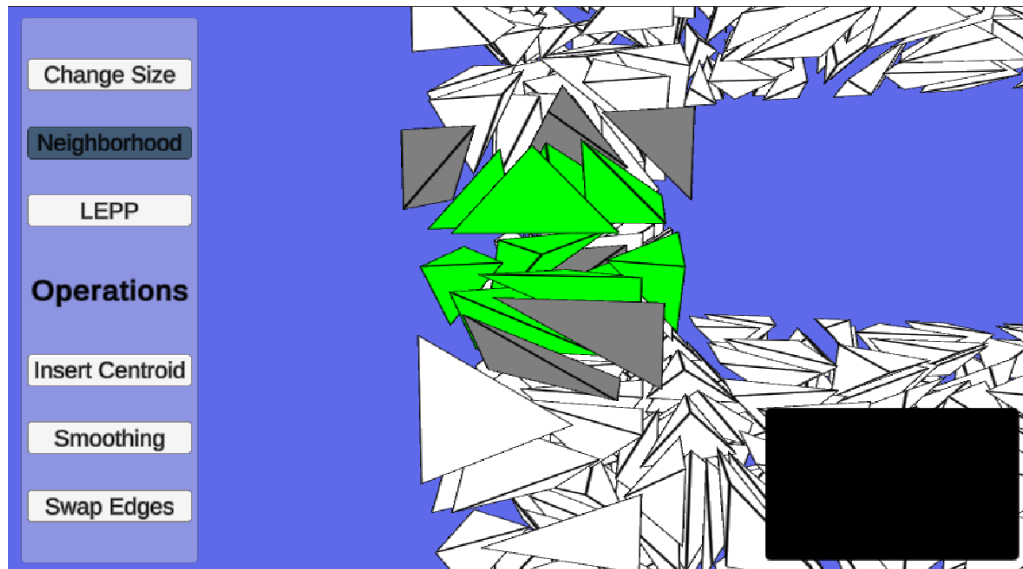


Figura 5.14: Vecinos de un conjunto de tetraedros.

# Capítulo 6

## Evaluación

### 6.1. Rendimiento

Para evaluar el rendimiento de la aplicación, se utilizó un computador con las siguientes características:

- Sistema operativo Windows 11 64 bits.
- Procesador AMD Ryzen 7 6800H 3.2 GHz, 8 procesadores principales, 16 procesadores lógicos.
- 16 GB de memoria RAM.
- Tarjeta Gráfica NVIDIA GeForce RTX 3060 Laptop GPU con 6 GB de GDDR6 y 3840 CUDA cores.

El desempeño que tiene la aplicación a la hora de dibujar y mover la malla depende de la cantidad de objetos que estén dentro del campo de visión, ya que estos son los que se renderizan. Para evaluar el desempeño que tiene la aplicación en esta tarea se utilizaron distintos modelos (incluidos en la aplicación) siendo mostrados en su totalidad en cámara.

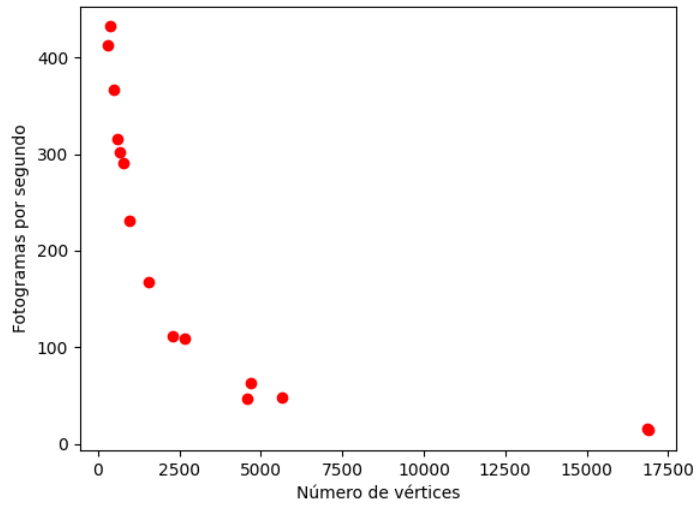


Figura 6.1: Rendimiento en función de la cantidad de vértices del modelo.

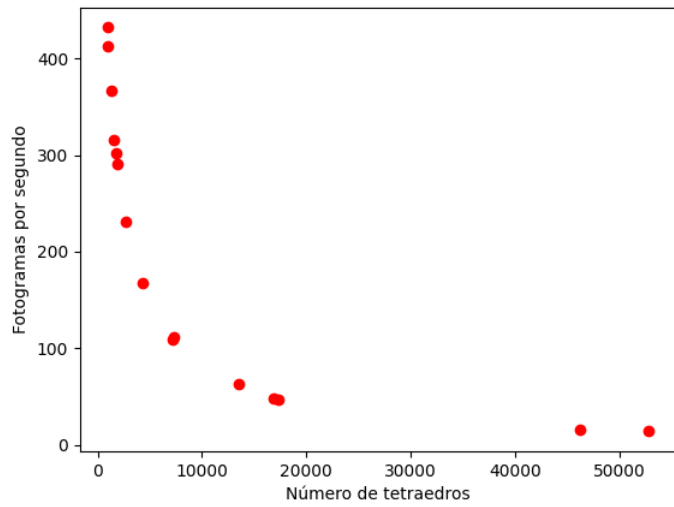


Figura 6.2: Rendimiento en función de la cantidad de tetraedros del modelo.

Se puede observar que tanto en la Figura 6.1 como en la Figura 6.2, el rendimiento disminuye cuando la cantidad de vértices y tetraedros aumenta.

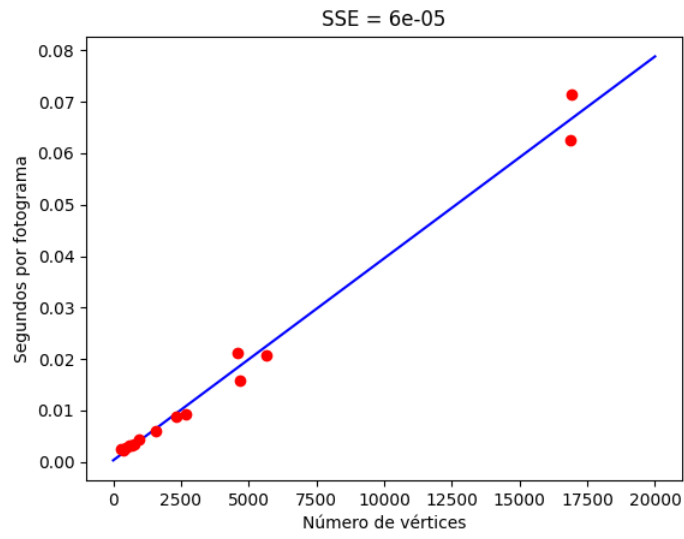


Figura 6.3: Tiempo de cada fotograma en función de la cantidad de vértices.

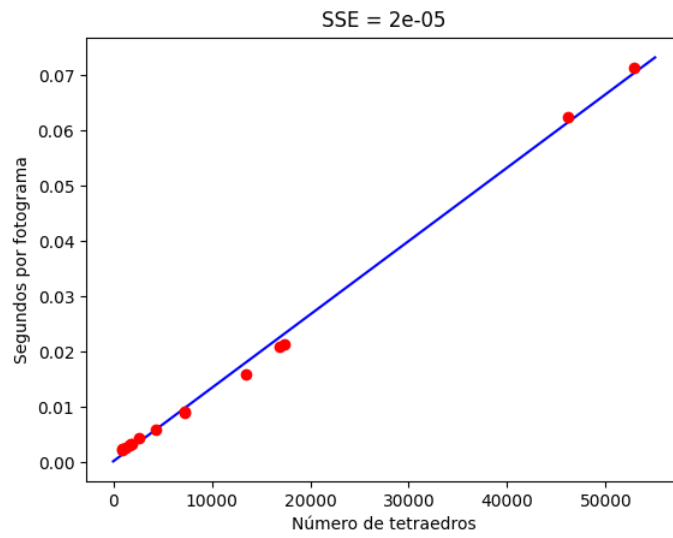


Figura 6.4: Tiempo de cada fotograma en función de la cantidad de tetraedros.



Si observamos el tiempo que se demora la aplicación en renderizar cada fotograma, podemos observar un comportamiento lineal. Más aún, en las Figuras 6.3 y 6.4 se puede apreciar que se pueden modelar los datos con una regresión lineal, obteniendo una suma de errores cuadrados (SSE) de 0.00006 y 0.00002 respectivamente.

En el caso del algoritmo LEPP, el rendimiento está más ligado a la cantidad de tetraedros que se recorren y no tanto a la cantidad de tetraedros del modelo.

Modelo	Vértices	Tetraedros	FPS	Iteraciones	Tetraedros visitados	Milisegundos
P	289	926	413	1	4	0.0548
P	289	926	413	3	13	0.1055
P	289	926	413	4	13	0.1432
Piece	4571	17319	47	1	3	0.0517
Piece	4571	17319	47	3	12	0.1052
Piece	4571	17319	47	4	18	0.1902
Triceratops	16854	46202	16	4	16	0.1382
Triceratops	16854	46202	16	6	30	0.2074
Triceratops	16854	46202	16	9	47	0.3715

Figura 6.5: Rendimiento del algoritmo simple LEPP.

El rendimiento del algoritmo simple LEPP aumenta en función de las iteraciones y cantidad de tetraedros visitados, sin embargo, los tiempos de espera son menores a 1 milisegundo en los casos más demandantes que pueden ocurrir en la herramienta (Figura 6.5). Este desempeño es adecuado para una respuesta fluida de la aplicación.

Cabe mencionar, además, que cada ciclo de iteración del algoritmo es pintado de manera asíncrona, por lo que independiente de la velocidad de la operación, los grupos de tetraedros son pintados a una velocidad moderada y constante.

Modelo	Vértices	Tetraedros	FPS	Tamaño estrella terminal	Milisegundos
P	289	926	413	2	0.6017
P	289	926	413	3	0.6932
P	289	926	413	5	1.1035
Piece	4571	17319	47	3	1.6844
Piece	4571	17319	47	4	2.6993
Piece	4571	17319	47	6	2.8342
Triceratops	16854	46202	16	4	2.9145
Triceratops	16854	46202	16	5	4.7203
Triceratops	16854	46202	16	9	8.7799

Figura 6.6: Rendimiento de la operación de inserción.

Modelo	Vértices	Tetraedros	FPS	Cantidad de tetraedros	Milisegundos
P	289	926	413	15	1.2741
P	289	926	413	20	1.9029
P	289	926	413	28	2.3052
Piece	4571	17319	47	24	6.4432
Piece	4571	17319	47	32	7.8945
Piece	4571	17319	47	48	8.1005
Triceratops	16854	46202	16	54	32.8381
Triceratops	16854	46202	16	70	38.6729
Triceratops	16854	46202	16	110	71.5559

Figura 6.7: Rendimiento de la operación de suavizamiento.

Al igual que antes, el tiempo de las operaciones de inserción y suavizamiento están ligadas

a la cantidad de tetraedros que son objetivo directo de las funciones (Figura 6.6 y 6.7). Si bien, estas operaciones son más demandantes que el simple LEPP, siguen siendo del orden de milisegundos en los peores casos.

El rendimiento de la función de intercambio de aristas es constante, ya que es una versión extremadamente restringida al aceptar solo 2 tipos de estrellas terminales.

De los resultados obtenidos, vemos que el rendimiento está ligado a la cantidad de datos del modelo y, aunque es posible que la herramienta tenga mal rendimiento con modelos extremadamente grandes, es una herramienta de visualización lo suficientemente poderosa como para poder renderizar y operar sobre mallas reales de manera interactiva en tiempo real.

## 6.2. Validación

La herramienta es capaz de leer mallas geométricas que han sido generadas previamente en otros programas, estas mallas como las que ya posee la aplicación son triangulaciones válidas del conjunto de puntos del modelo.

Teniendo esto en cuenta, mantener un mallado correcto del modelo es algo crucial para que la aplicación funcione correctamente. Al realizar modificaciones a la malla 3D válida que se tiene como entrada, hay que asegurar que la triangulación resultante sean correctas.

Para esto se realizan distintas revisiones a lo largo de cada operación

- Verificación de vecinos, asegurándose que cada tetraedro se relacione con un tetraedro por cara, a excepción de los tetraedros bordes.
- Revisión de los cálculos luego de realizada la operación, verificando que vértices y tetraedros resulten válidos.
- Verificación de los nuevos tetraedros y que estos reemplacen correctamente la estrella terminal.

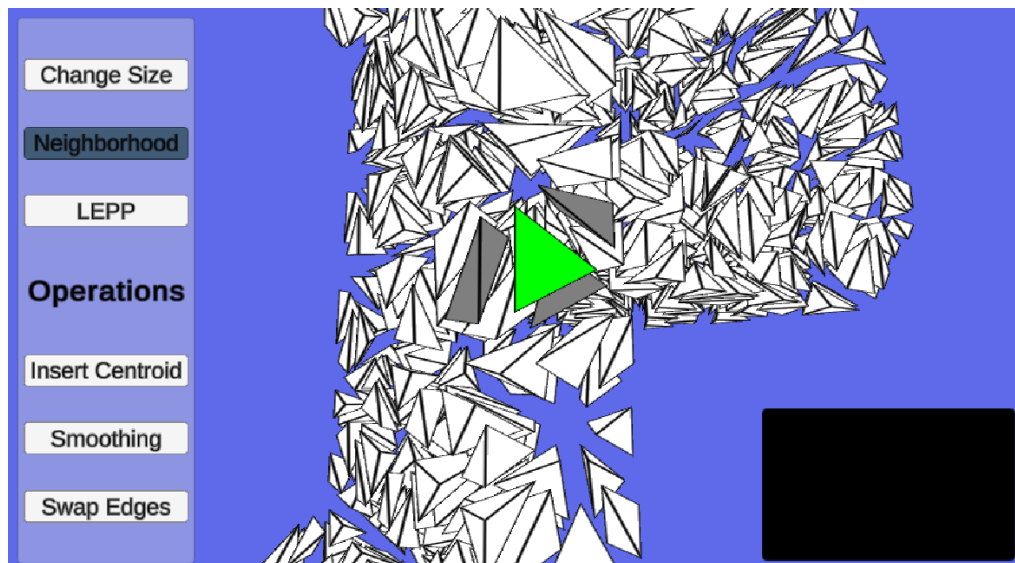


Figura 6.8: Vecinos del tetraedro original de la malla.

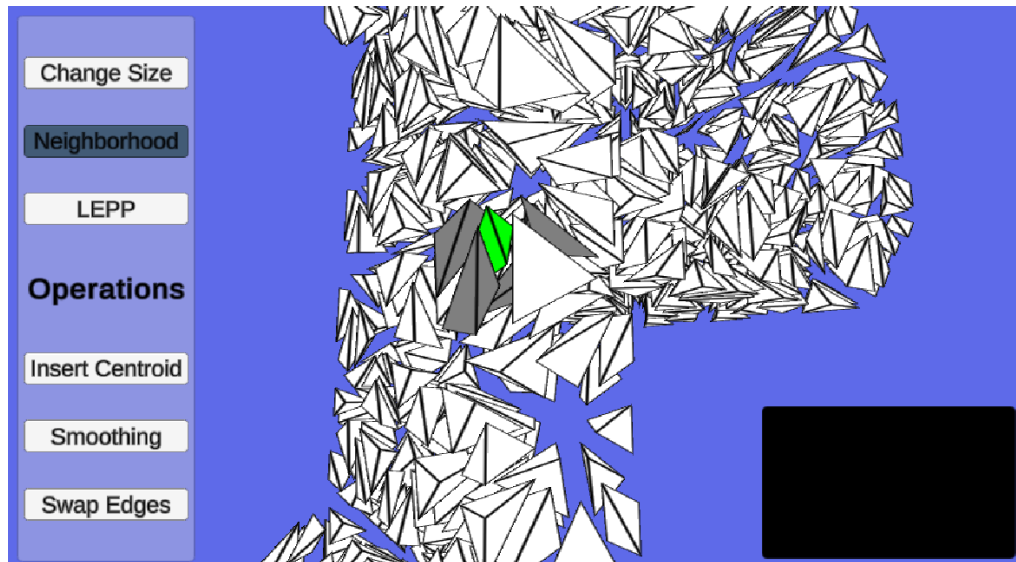


Figura 6.9: vecinos del tetraedro nuevo de la malla modificada.

Además, la herramienta tiene la capacidad de chequear la correctitud de la malla (ya sea modificada o no), chequeando que cada tetraedro de esta cumpla con los requisitos para ser uno (como el número de caras, vértices y aristas). También se valida que los vecinos sean válidos y pertenezcan a la malla.

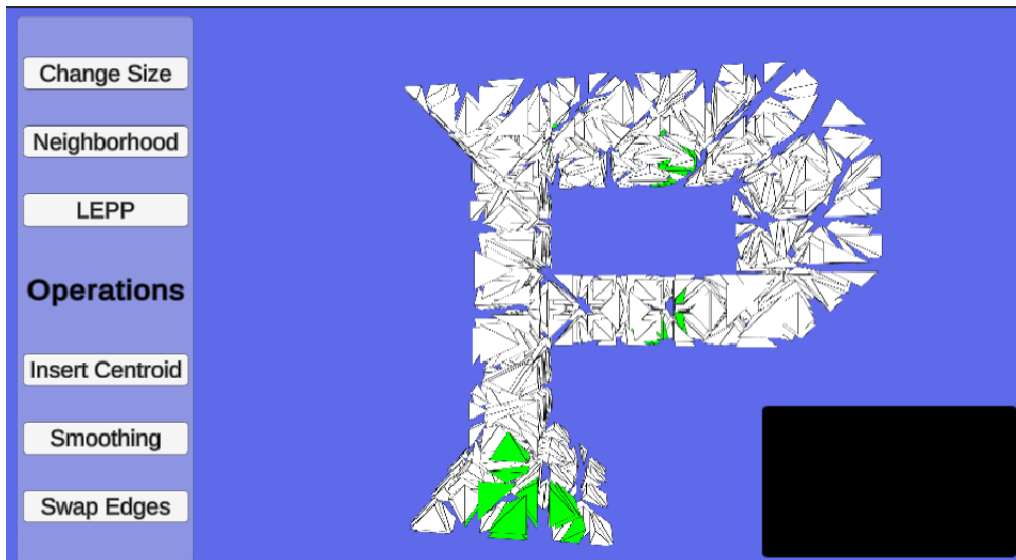


Figura 6.10: Validación de los primero 30 tetraedros de la malla original.

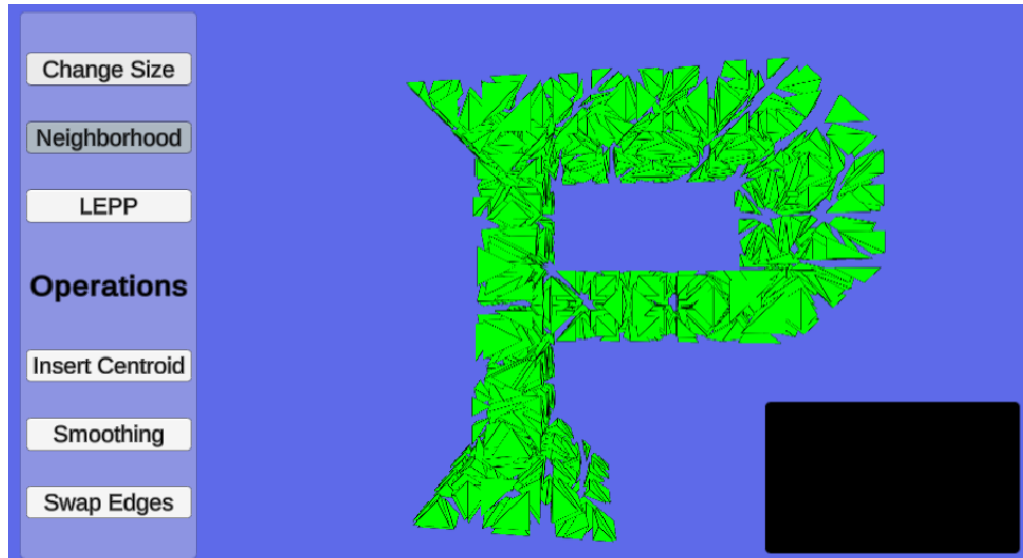


Figura 6.11: Validación de la malla original.

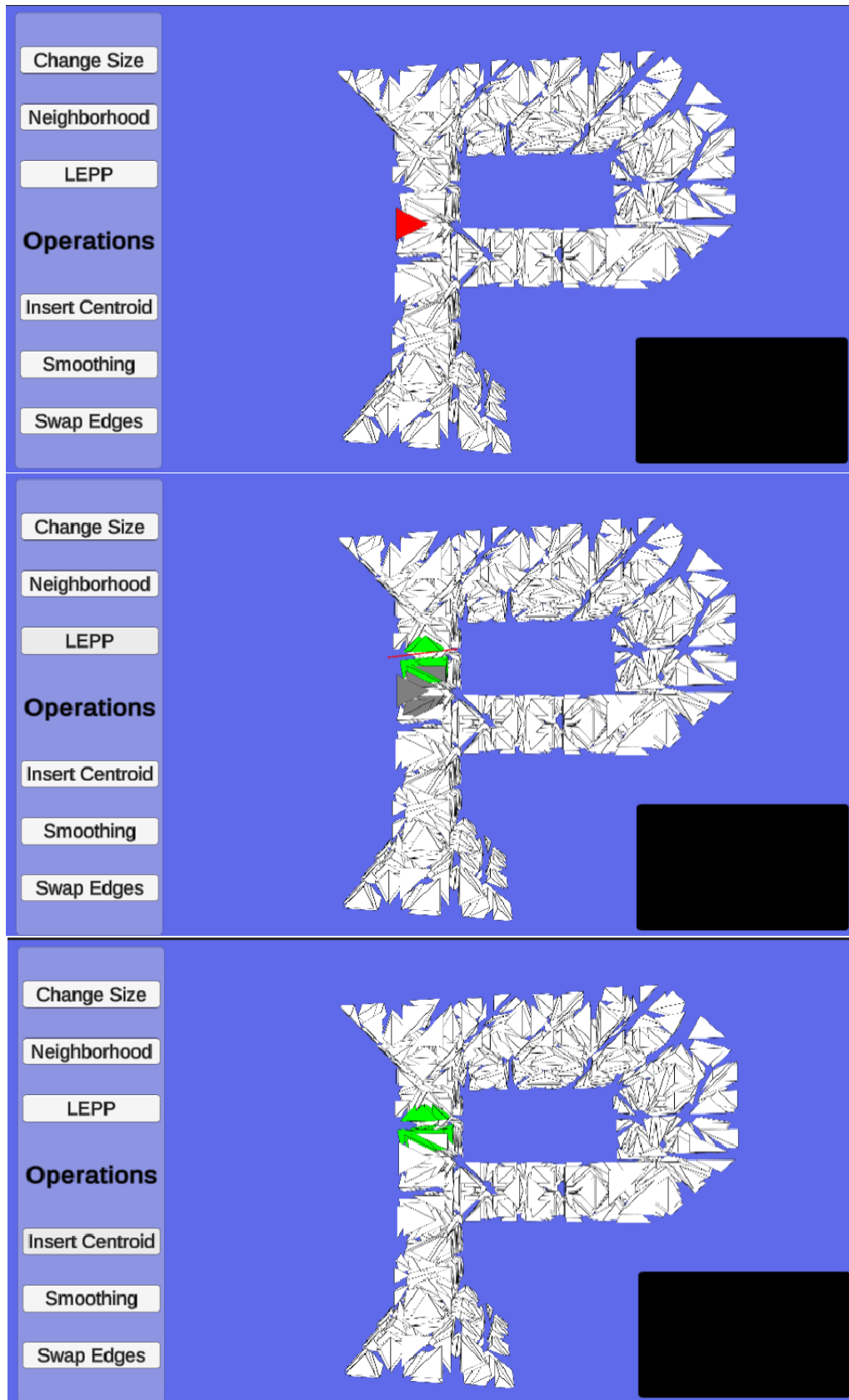


Figura 6.12: Inserción de centroide en la malla.

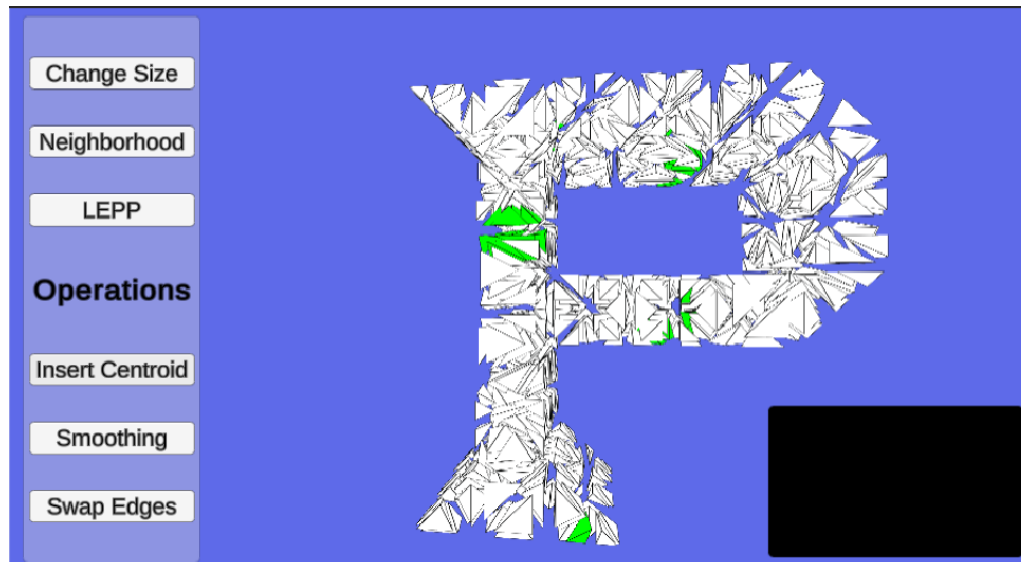


Figura 6.13: Validación de los primeros 30 tetraedros de la malla modificada.

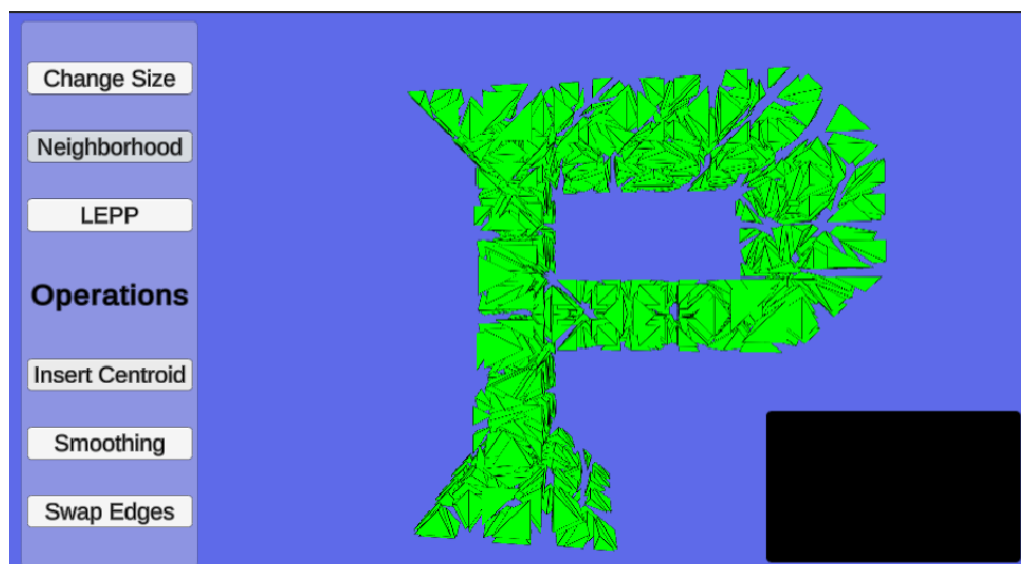


Figura 6.14: Validación de la malla modificada.

# Capítulo 7

## Conclusiones

La herramienta para la visualización y operación sobre mallas de tetraedros ha resultado ser una herramienta valiosa y poderosa para la exploración y comprensión de estas mallas volumétricas.

A lo largo de este trabajo hemos visto que los objetivos que tenía la creación de esta herramienta se cumplen, se logró un correcto y buen estructuramiento de los datos para almacenar la malla de una manera eficiente. Con esto, también se logró una correcta visualización del objeto modelado.

Se ha mencionado como una interfaz gráfica intuitiva y eficiente, permite a los usuarios interactuar con las mallas de tetraedros de una manera efectiva y así obtener información valiosa a partir de sus visualizaciones. La aplicación logra esto, dándole a los usuarios las herramientas necesarias para manipular la malla a través de distintos botones, además de permitir moverse por el espacio de la escena para lograr ver de distintos ángulos la renderización.

La aplicación logra ser capaz de realizar diversas operaciones en mallas, como por ejemplo el suavizado y la inserción de puntos. Con esto los usuarios son capaces de visualizar de una manera sencilla como estas operaciones afectan la malla volumétrica.

También, se proveyó una manera simple de poder verificar la validez de la malla luego de ser modificada por las operaciones solicitadas por el usuario.

Con todos los objetivos anteriores cumplidos se concluye que estas herramientas poseen un gran potencial para el área, promoviendo el estudio de las mallas geométricas y volumétricas para generar futuras investigaciones, sin contar que de por sí pueden ser de gran utilidad en estas.

### 7.1. Limitaciones

Si bien se lograron muchas cosas en este trabajo, existen muchas cosas que fueron descartadas o simplificadas en función del tiempo acotado y la complejidad del tema tratado. Entre las limitaciones que se encuentran en esta herramienta está la exclusividad en el formato de archivos aceptados, ya que si el archivo no está en forma .m3d no va a poder ser cargado por la aplicación.

Otra limitación es la baja cantidad de operaciones implementadas debido a la complejidad de estas y a la dificultad de trabajar en espacios 3D. Cabe destacar que también se consideraron limitaciones de cálculo, ya que algunos algoritmos (como la versión completa de LEPP) suponen una gran cantidad de cómputo en mallas particularmente complejas.

Como se dió a entender en el capítulo anterior, a medida que la malla de tetraedros aumenta en tamaño, la aplicación se enfrenta a problemas de rendimiento. La visualización y manipulación de mallas muy grandes podría requerir una cantidad significativa de recursos del sistema, lo que podría afectar la experiencia del usuario.

La compatibilidad también es una restricción del proyecto, ya requiere de harto cómputo es posible que no pueda ser utilizado en todas las máquinas. Además, esta herramienta está pensada solo para ser utilizada en computadores.

## 7.2. Sugerencias y futuros trabajos

A partir de este estudio, se abren diversas oportunidades para trabajos futuros. Partiendo con solucionar las limitaciones existentes.

- Una mejora importante podría centrarse en la optimización del rendimiento de la aplicación. Con, por ejemplo, algoritmos más eficientes para manejar mallas más grandes y complejas sin comprometer la fluidez y la capacidad de respuesta de la interfaz
- Ampliar la compatibilidad de la aplicación para trabajar con una variedad de formatos de archivos de mallas 3D. Esto permitiría a los usuarios importar y exportar mallas en distintos formatos.
- Implementar más operaciones para modificar la malla de tetraedros podría ampliar las capacidades de la aplicación. Logrando así el aprendizaje de otros algoritmos.
- Mejorar la interfaz gráfica para hacerla aún más fácil y cómoda de usar.
- Mejorar la completitud de la estructura de datos, permitiendo usos más complejos y eficientes de los datos.
- Aumentar las maneras de validar de las mallas modificadas durante el uso del usuario.

También se es posible extender este trabajo de las siguientes maneras:

- Integración con realidad virtual permitiendo así una forma totalmente inmersiva de explorar la malla volumétrica.
- Agregar funcionalidades de almacenamiento en la nube, permitiendo así poder cargar datos desde cualquier parte a la aplicación.
- Inclusión de un sistema de colaboración en línea para que de esta manera múltiples usuarios puedan experimentar los mismos efectos en las mallas.



# Bibliografía

- [1] Rodríguez Moreno, P. Á., “Parallel lepp-based algorithms for the generation and refinement of triangulations,” 2015, <https://repositorio.uchile.cl/handle/2250/135082>.
- [2] Rodriguez, P. A. y Rivara, M.-C., “Multithread lepp-bisection algorithm for tetrahedral meshes,” en Proceedings of the 22nd International Meshing Roundtable, 2014, [https://doi.org/10.1007/978-3-319-02335-9\\_29](https://doi.org/10.1007/978-3-319-02335-9_29).
- [3] Wikipedia contributors, “Delaunay triangulation — Wikipedia, the free encyclopedia,” 2022, [https://en.wikipedia.org/w/index.php?title=Delaunay\\_triangulation&oldid=1111405984](https://en.wikipedia.org/w/index.php?title=Delaunay_triangulation&oldid=1111405984).
- [4] Maur, P., “Delaunay triangulation in 3d,” Technical Report, Departmen. of Computer Science and Engineering, 2002.
- [5] Unity Technologies, “Unity.”, <https://unity.com>.
- [6] Balboa, F., Rodríguez-Moreno, P. A., y Rivara, M. C., “Terminal star operations algorithm for tetrahedral mesh improvement,” en 27th International Meshing Roundtable, IMR 2018, Albuquerque, NM, USA, October 1-5, 2018, vol. 127 de Lecture Notes in Computational Science and Engineering, pp. 269–282, Springer, 2018, [doi:10.1007/978-3-030-13992-6\\_15](https://doi.org/10.1007/978-3-030-13992-6_15).
- [7] Fernando Balboa, P. R.-M. y M.C., R., “Terminal star operations algorithm for tetrahedral mesh improvement,” 2018, [doi:10.2307/2267552](https://doi.org/10.2307/2267552).
- [8] Rivara, M.-C., “New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations,” International Journal for Numerical Methods in Engineering, vol. 40, pp. 3313 – 3324, 1997, [doi:10.1002/\(SICI\)1097-0207\(19970930\)40:18<3313::AID-NME214>3.0.CO;2-#](https://doi.org/10.1002/(SICI)1097-0207(19970930)40:18<3313::AID-NME214>3.0.CO;2-#).
- [9] Balboa Verdugo, F. A., “Mejoramiento de mallas de tetraedros mediante algoritmos lepp,” 2020.
- [10] Wikipedia contributors, “Computer-aided design — Wikipedia, the free encyclopedia,” 2023, [https://en.wikipedia.org/w/index.php?title=Computer-aided\\_design&oldid=1166600711](https://en.wikipedia.org/w/index.php?title=Computer-aided_design&oldid=1166600711). [Online; accessed 24-July-2023].
- [11] Wikipedia contributors, “Meshlab — Wikipedia, the free encyclopedia,” 2022, <https://en.wikipedia.org/w/index.php?title=MeshLab&oldid=1115421547>.
- [12] Dassi, F., Kamenski, L., Farrell, P., y Si, H., “Tetrahedral mesh improvement using moving mesh smoothing, lazy searching flips, and rbf surface reconstruction,” Computer-Aided Design, vol. 103, pp. 2–13, 2018, [doi:https://doi.org/10.1016/j.cad.2017.11.010](https://doi.org/10.1016/j.cad.2017.11.010). 25th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

- [13] Freitag, L. A. y Ollivier-Gooch, C., “Tetrahedral mesh improvement using swapping and smoothing,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 21, pp. 3979–4002, 1997, doi:[https://doi.org/10.1002/\(SICI\)1097-0207\(19971115\)40:21<3979::AID-NME251>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9).