



Estructuras de datos

Proyecto

Entrega 0 y 1

Juan Sebastian Rodriguez Pabon

Julian Perez Gomes

David Rodriguez Jurado

Jan Marco Muñoz Pineda

Daniel Galvis Betancourth

Septiembre 06/09/2024

Documento de diseño.

Mediante un conjunto de vértices, aristas y caras permiten definir digitalmente la forma de un objeto tridimensional en un espacio 3D, para su correcta manipulación se conforma de algoritmos de análisis y procedimiento de mallas poligonales, a su misma vez, estas mallas son usadas para aplicaciones de simulación, videojuegos, animación, realidad virtual y aumentada, sistemas CAD (Computer-Aided- Design) y muchas otras.

La información de vértices, aristas y caras debe estructurarse y almacenarse de forma que facilite la inserción, búsqueda y eliminación de los datos de cada objeto 3D.

Entrada

La información almacenada se encuentra en un archivo de texto el cual tiene la siguiente estructura:

```
mesh_name
n
x0 y0 z0
...
xn-1 yn-1 zn-1
c1 i1_1 i1_2 ... i1_c
...
cm im_1 im_2 ... im_c
-1
```

Siendo de tal manera que:

- mesh_name; Es una cadena de caracteres que representa el nombre de la malla u objeto 3D (sin espacios).
- n: Es un numero entero, es la cantidad de puntos (vértices) que definen la malla, es decir

cuántos vértices tiene la malla en el espacio 3D.

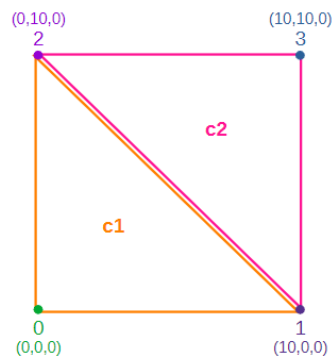
- x_i, y_i y z_i : Son las coordenadas reales del i -ésimo punto (vértice), es decir cada punto (vértice) de la figura tiene 3 coordenadas x, y, z . Hasta la línea de X_{n-1} son las coordenadas de cada vértice.

- $c_1 \ i1_1 \ i1_2 \dots i1_c$ hasta $c_m \ im_1 \ im_2 \dots im_c$ Describe las conexiones entre los vértices.
 $i1_1 \ i1_2 \dots i1_c \ c1$: El índice de los vértices que forman la cara o arista

índices de los vértices que forman la cara o arista

- c_j es el tamaño (cantidad de puntos o vértices) de la j -ésima cara.
- ij_k es el índice del k -ésimo punto (vértice) que conforma la j -ésima cara. Note que los puntos (vértices) están indexados desde 0 hasta $n-1$, en el orden de aparición en el archivo.
- -1 indica el final de la información de la malla.

A continuación, se presenta un ejemplo de malla (dos triángulos), y al lado la información correspondiente en el archivo de texto:



```
Mesh_0
4
0 0 0
10 0 0
0 10 0
10 10 0
3 0 1 2
3 1 3 2
-1
```

En dicha imagen se puede observar la Implementación de un archivo mencionado anteriormente, en dicho archivo primero se tiene Mesh_0 el cual nos indica el nombre de la malla, en este caso seria. Después las siguientes 4 líneas se tienen las coordenadas de cada vértice, cada vértice tiene 3 coordenadas. Los otros dos comandos describen las caras de la figura, es decir, el primero número (3) indica la cantidad de vértices que tiene esa parte de la figura, en este caso un triángulo, los siguientes 3 números son los índices de los vértices que forman esa cara.

Componentes

El objetivo del proyecto es construir un sistema que permita realizar operaciones sencillas de mallas poligonales recibiendo comandos textuales con funcionalidades específicas.

Objetivo:

Los algoritmos de este componente gestionan la información básica de objetos 3D a partir de archivos de mallas. El propósito es cargar, almacenar y estructurar eficientemente la información para facilitar su recuperación posterior.

Comando: *cargar nombre_archivo*

Descripción:

Este comando carga la Información del objeto 3D desde un archivo de malla (nombre_archivo). El archivo contiene la representación de los vértices y las conexiones entre ellos, permitiendo así la creación de un objeto tridimensional que será almacenado en memoria.

Entradas:

- **nombre_archivo:** Es una cadena de caracteres que representa el nombre del archivo de malla que se desea cargar. Este archivo contiene las coordenadas de los vértices y las caras del objeto 3D.

Salidas:

- **(Archivo vacío o incompleto):** Si el archivo no contiene una estructura válida para representar un objeto 3D, se debe mostrar este mensaje.
- **(Archivo no existe):** Si el archivo no se encuentra o no se puede leer, se muestra este error.
- **(Objeto ya existe):** Si el objeto con el mismo nombre ya fue cargado previamente, este mensaje alerta al usuario de que no se puede cargar de nuevo.
- **(Resultado exitoso):** Si todo está en orden, el sistema confirma que el objeto ha sido cargado correctamente en memoria.

Comando:Listado

Entradas:

- No requiere entradas.

Salidas:

- **(Memoria vacía):** Si no hay ningún objeto cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra un listado de los objetos cargados en memoria con la siguiente información:
 - Nombre de cada objeto.
 - Cantidad de vértices que tiene cada objeto (n).
 - Cantidad de aristas (a).
 - Cantidad de caras (c).

Comando: envolver nombre_objeto

Entradas:

- **nombre_objeto:** Es una cadena de caracteres que representa el nombre del objeto del cual se desea calcular la caja envolvente.

Salidas:

- **(Objeto no existe):** Si el objeto especificado no ha sido cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se genera la caja envolvente del objeto, se le asigna un nombre automáticamente (env_nombre_objeto), y se agrega a los objetos en memoria.

Comando: envolver

Entradas:

- No requiere entradas.

Salidas:

- **(Memoria vacía):** Si no hay ningún objeto cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se genera la caja envolvente que incluye todos los objetos en memoria. Esta caja recibe el nombre **env_global** y se agrega a los objetos en memoria.

Comando: descargar nombre_objeto

Entradas:

- **nombre_objeto:** Es una cadena de caracteres que representa el nombre del objeto que se desea eliminar de la memoria.

Salidas:

- **(Objeto no existe):** Si el objeto especificado no está cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** El objeto **nombre_objeto** ha sido eliminado correctamente de la memoria.

Comando: guardar nombre_objeto nombre_archivo

Entradas:

- **nombre_objeto:** Es el nombre del objeto en memoria que se desea guardar.
- **nombre_archivo:** Es el nombre del archivo donde se guardará la información del objeto.

Salidas:

- **(Objeto no existe):** Si el objeto **nombre_objeto** no está en memoria, se muestra este mensaje.
- **(Resultado exitoso):** La información del objeto **nombre_objeto** se ha guardado correctamente en el archivo **nombre_archivo**.

Comando: salir

Entradas:

- No requiere entradas.

Salidas:

- **(No tiene salida por pantalla):** Este comando no produce ninguna salida en pantalla.

Componente 2

Comando: v_cercano px py pz nombre_objeto

Entradas:

- **px, py, pz:** Coordenadas del punto en el espacio tridimensional al que se quiere encontrar el vértice más cercano.
- **nombre_objeto:** Nombre del objeto cargado en memoria para el cual se busca el vértice más cercano.

Salidas:

- **(Objeto no existe):** Si el objeto **nombre_objeto** no está cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra el índice del vértice más cercano **i** y sus coordenadas **(vx, vy, vz)**, junto con la distancia euclidiana entre el vértice y el punto **(px, py, pz)**.

Comando: v_cercano px py pz

Entradas:

- **px, py, pz:** Coordenadas del punto en el espacio tridimensional al que se quiere encontrar el vértice más cercano, entre todos los objetos cargados en memoria.

Salidas:

- **(Memoria vacía):** Si no hay ningún objeto cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra el objeto al que pertenece el vértice más cercano, el índice del vértice **i**, sus coordenadas (**vx, vy, vz**), y la distancia euclidiana entre el vértice y el punto (**px, py, pz**).

Comando: **v_cercanos_caja nombre_objeto**

Entradas:

- **nombre_objeto:** Nombre del objeto cargado en memoria del cual se quiere encontrar los vértices más cercanos a las esquinas de su caja envolvente.

Salidas:

- **(Objeto no existe):** Si el objeto **nombre_objeto** no está cargado en memoria, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra una tabla con las coordenadas de cada esquina de la caja envolvente (**e1x, e1y, e1z**), (**e2x, e2y, e2z**), ..., (**e8x, e8y, e8z**), el índice del vértice más cercano a cada esquina **i1, i2, ..., i8**, sus coordenadas (**v1x, v1y, v1z**), (**v2x, v2y, v2z**), ..., (**v8x, v8y, v8z**), y la distancia euclidiana entre cada vértice y su respectiva esquina **valor_distancia_1, valor_distancia_2, ..., valor_distancia_8**.

Componente 3

Comando: **ruta_corta i1 i2 nombre_objeto**

Entradas:

- **i1, i2:** Índices de los vértices que se desea conectar.
- **nombre_objeto:** Nombre del objeto cargado en memoria para el cual se desea calcular la ruta más corta entre los vértices.

Salidas:

- **(Objeto no existe):** Si el objeto **nombre_objeto** no está cargado en memoria, se muestra este mensaje.
- **(Índices iguales):** Si los índices de los vértices dados son iguales, se muestra este mensaje.
- **(Índices no existen):** Si los índices de los vértices están fuera del rango válido del objeto, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra la ruta más corta que conecta los vértices **i1** y **i2**, especificando los vértices intermedios por los que pasa **i1, v1, v2, ..., vn, i2**, junto con la longitud total de la ruta **valor_distancia**.

Comando: ruta_corta_centro i1 nombre_objeto

Entradas:

- **i1:** Índice del vértice desde el cual se desea calcular la ruta más corta hacia el centro del objeto.
- **nombre_objeto:** Nombre del objeto cargado en memoria para el cual se desea calcular la ruta más corta hacia su centro.

Salidas:

- **(Objeto no existe):** Si el objeto **nombre_objeto** no está cargado en memoria, se muestra este mensaje.
- **(Índice no existe):** Si el índice del vértice **i1** está fuera del rango válido del objeto, se muestra este mensaje.
- **(Resultado exitoso):** Se muestra la ruta más corta que conecta el vértice **i1** con el centroide del objeto, identificado por las coordenadas **ct (ctx, cty, ctz)**, y los vértices intermedios **i1, v1, v2, ..., ct**, junto con la longitud total de la ruta **valor_distancia**.

TADS

TAD 1: *Vértice*

Descripción:

Representa un punto en el espacio 3D con coordenadas $\backslash(x\backslash)$, $\backslash(y\backslash)$ y $\backslash(z\backslash)$.

Operaciones:

- *CrearVértice(x, y, z)*: Crea un vértice con las coordenadas proporcionadas.
- *ObtenerCoordenadas(vértice)*: Retorna las coordenadas $\backslash(x\backslash)$, $\backslash(y\backslash)$, $\backslash(z\backslash)$ del vértice.
- *ModificarCoordenadas(vértice, x, y, z)*: Modifica las coordenadas del vértice.

Entradas:

- $\backslash(x\backslash)$, $\backslash(y\backslash)$, $\backslash(z\backslash)$: Coordenadas reales de un punto en el espacio 3D.

Salidas:

- Un nuevo vértice con las coordenadas dadas.

TAD 2: *Arista*

Descripción:

Representa una arista entre dos vértices en una malla poligonal.

Operaciones:

- *CrearArista(vértice1, vértice2)*: Crea una arista entre dos vértices.
- *ObtenerVértices(arista)*: Retorna los vértices que forman la arista.
- *ModificarArista(arista, vértice1, vértice2)*: Cambia los vértices que forman la arista.

Entradas:

- $\backslash(\text{vértice1}\backslash)$, $\backslash(\text{vértice2}\backslash)$: Dos vértices que forman la arista.

Salidas:

- Una arista que conecta los dos vértices.

TAD 3: *Cara*

Descripción:

Representa una cara en una malla 3D, formada por un conjunto de vértices.

Operaciones:

- *CrearCara(vértices)*: Crea una cara con los vértices dados.
- *ObtenerVérticesCara(cara)*: Retorna la lista de vértices que forman la cara.
- *AñadirVértice(cara, vértice)*: Añade un vértice a la cara.
- *EliminarVértice(cara, vértice)*: Elimina un vértice de la cara.

Entradas:

- \(\text{vértices}\): Lista de vértices que forman la cara.

Salidas:

- Una nueva cara formada por los vértices dados.

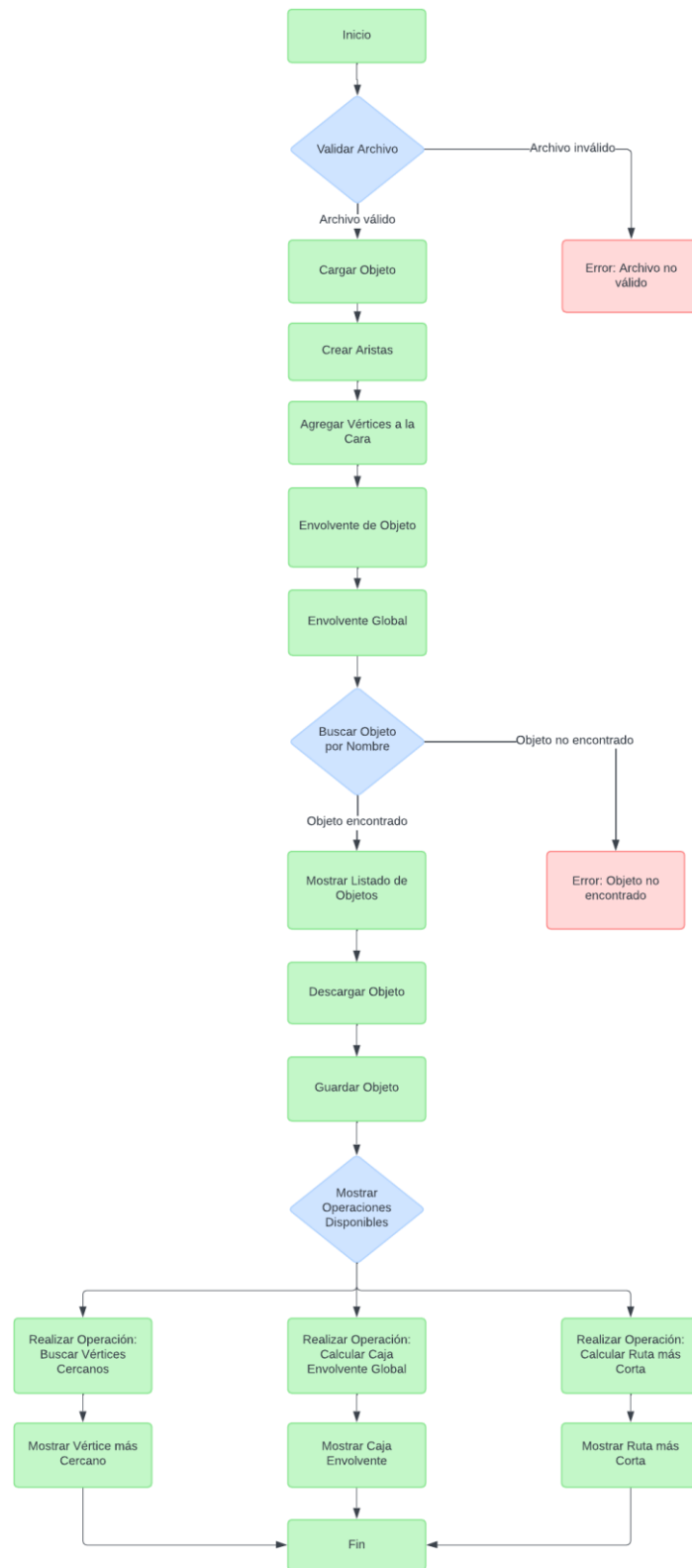
Estos TADs proporcionan una estructura básica para almacenar y manipular los vértices, aristas y caras en una malla poligonal 3D. Para implementar estos conceptos, sería necesario también definir las interacciones entre los TADs en el sistema de comandos.

- **Diagramas y gráficos**

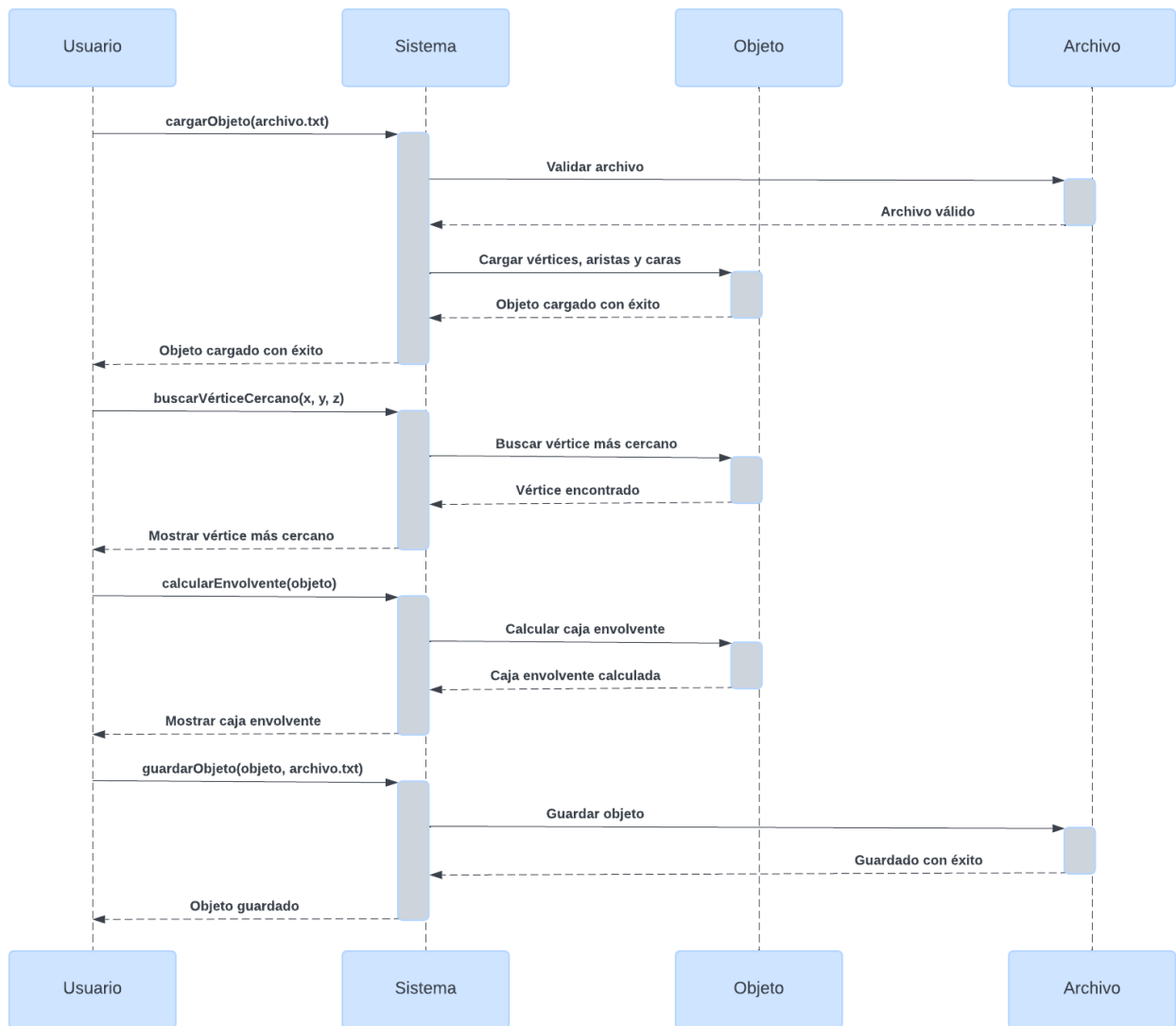
1. Diagrama de relación.



2. Diagrama de flujo.



3. Diagrama de secuencia.



Plan de pruebas

Entrada archivo	Salida esperada	Salida obtenida
Mesh_0	Nombre de malla: Mesh_0	Nombre de malla: Mesh_0
4	Número de vértices: 4	Número de vértices: 4
0 0 0	Coordenada vértice 0: (0, 0, 0)	Coordenada vértice 0: (0, 0, 0)
10 0 0	Coordenada vértice 1: (10, 0, 0)	Coordenada vértice 1: (10, 0, 0)
0 10 0	Coordenada vértice 2: (0, 10, 0)	Coordenada vértice 2: (0, 10, 0)
10 10 0	Coordenada vértice 3: (10, 10, 0)	Coordenada vértice 3: (10, 10, 0)
3 0 1 2	Cara 1 formada por vértices 0, 1 y 2	Cara 1 formada por vértices 0, 1 y 2
3 1 3 2	Cara 2 formada por vértices 1, 3, 2	Cara 2 formada por vértices 1, 3, 2
-1	Fin de archivo.	Fin de archivo.
Comando: Cargar archivo	El objeto objeto.getNombre() ha sido cargado exitosamente desde el archivo "ruta".	El objeto objeto.getNombre() ha sido cargado exitosamente desde el archivo "ruta".
Comando: Listado	Mesh_0 contiene 4 vértices, 5 aristas y 2 caras.	Mesh_0 contiene 4 vértices, 5 aristas y 2 caras.
Comando: Envolvente Mesh_0	La caja envolvente del objeto Mesh_0 se ha generado con el nombre env_Mesh_0 y se ha agregado a los objetos en memoria. (Información de los vértices y caras de la caja envolvente)	Vertices: (0, 0, 0) (10, 0, 0) (10, 10, 0) (0, 10, 0) (0, 0, 0) (10, 0, 0) (10, 10, 0) (0, 10, 0) Caras: Cara frontal: 0, 1, 2, 3 Cara trasera: 4, 5, 6, 7 Cara izquierda: 0, 4, 7, 3 Cara derecha: 1, 5, 6, 2 Cara superior: 3, 2, 6, 7 Cara inferior: 0, 1, 5, 4
Comando: Envolvente (global)	La caja envolvente de los objetos en memoria se ha	Mesh_0

	<p>generado con el nombre env_global y se ha agregado a los objetos en memoria. (Información de los vértices y caras de las cajas envolventes)</p>	<p>Vertices: (0, 0, 0) (10, 0, 0) (10, 10, 0) (0, 10, 0) (0, 0, 0) (10, 0, 0) (10, 10, 0) (0, 10, 0) Caras: Cara frontal: 0, 1, 2, 3 Cara trasera: 4, 5, 6, 7 Cara izquierda: 0, 4, 7, 3 Cara derecha: 1, 5, 6, 2 Cara superior: 3, 2, 6, 7 Cara inferior: 0, 1, 5, 4</p>
Comando: Descargar	El objeto Mesh_0 ha sido eliminado de la memoria de trabajo.	El objeto Mesh_0 ha sido eliminado de la memoria de trabajo.
Comando: Ayuda (general)	<p>cargar nombre_archivo: Carga un objeto 3D desde un archivo.</p> <p>listado: Lista los objetos cargados en memoria.</p> <p>envolvente nombre_objeto: Calcula la caja envolvente de un objeto.</p> <p>envolvente: Calcula la caja envolvente global de todos los objetos.</p> <p>descargar nombre_objeto: Elimina un objeto de la memoria.</p> <p>guardar nombre_objeto nombre_archivo: Guarda un objeto en un archivo.</p>	<p>cargar nombre_archivo: Carga un objeto 3D desde un archivo.</p> <p>listado: Lista los objetos cargados en memoria.</p> <p>envolvente nombre_objeto: Calcula la caja envolvente de un objeto.</p> <p>envolvente: Calcula la caja envolvente global de todos los objetos.</p> <p>descargar nombre_objeto: Elimina un objeto de la memoria.</p> <p>guardar nombre_objeto nombre_archivo: Guarda un objeto en un archivo.</p>

	<p>salir: Termina la ejecución de la aplicación.</p> <p>ayuda: Muestra esta ayuda.</p> <p>ayuda comando: Muestra la ayuda para un comando específico.</p>	<p>salir: Termina la ejecución de la aplicación.</p> <p>ayuda: Muestra esta ayuda.</p> <p>ayuda comando: Muestra la ayuda para un comando específico.</p>
Comando: Ayuda “Cargar” (Ayuda para comando específico)	<p>Uso: cargar nombre_archivo</p> <p>Carga un objeto 3D desde un archivo.</p>	<p>Uso: cargar nombre_archivo</p> <p>Carga un objeto 3D desde un archivo.</p>
Comando: Salir	Termina la ejecución de la aplicación correctamente.	Termina la ejecución de la aplicación correctamente.
Comando: Guardar	La información del objeto Mesh_0 ha sido guardada exitosamente en el archivo salida0.txt.	La información del objeto Mesh_0 ha sido guardada exitosamente en el archivo salida0.txt.