



### **Entrega 3 Proyecto Estructura de Datos**

Isaías Acosta Herrera

Juan Diego Carreño Vasquez

Julian Diaz

Santiago Hernandez Rendon

Pontificia Universidad Javeriana de Colombia

John Corredor

Bogotá, Colombia

18 de Noviembre de 2024

## Componente 3

### Acta de evaluación de entregas anteriores

En la Entrega 1, obtuvimos una calificación de 4.2. Aunque no recibimos retroalimentación directa del profesor, seguimos las pautas indicadas y entregamos un código funcional junto con un documento bien organizado y los respectivos planes de prueba. Aun así, creemos que podríamos haber detallado un poco más el diseño y el funcionamiento del sistema para facilitar su comprensión..

En la Entrega 2, nuestra calificación fue 3.7. De nuevo, no hubo comentarios textuales, pero asumimos que el descenso en la nota pudo deberse a la falta de documentación en el código. En esta entrega final, corregimos incluyendo más documentación dentro del código, algo que probablemente no se resaltó lo suficiente en nuestro documento. Por eso, en esta última entrega hemos enfocado nuestros esfuerzos en mejorar la claridad del documento, describir más a fondo las funcionalidades nuevas y realizar correctamente nuestros planes de prueba

### Procedimiento Principal

El **Componente 3: rutas más cortas** se centra en encontrar los caminos más eficientes que conectan vértices dentro de los objetos 3D representados por mallas poligonales. Utilizando algoritmos que miden distancias euclidianas, este componente identifica rutas óptimas que unen diferentes puntos en la malla. La función `ruta_corta i1 i2 nombre_objeto` determina el camino más corto entre dos vértices específicos, proporcionando la secuencia de vértices que conforman la ruta y la distancia total. Por otro lado, la función `ruta_corta_centro i1 nombre_objeto` conecta un vértice dado con el centro del objeto (calculado como el centroide de sus vértices) mediante un camino mínimo, identificando el vértice más cercano al centroide y trazando una conexión eficiente. Ambas funciones son claves para manejar recorridos dentro de estructuras tridimensionales y facilitan el análisis de conectividad en mallas complejas.

### Comandos

- **Comando:** `ruta_corta i1 i2 nombre_objeto`

#### Entradas:

- `i1, i2`: Índices de los vértices para los que se desea encontrar la ruta más corta.
- `nombre_objeto`: Nombre del objeto 3D en el que se buscará la ruta.

#### Salidas:

- *Objeto no existe*: Mensaje que indica que el objeto nombre\_objeto no ha sido cargado en memoria.
- *Índices iguales*: Mensaje que informa que los índices de los vértices proporcionados son idénticos, por lo que no se requiere calcular la ruta.
- *Índices no existen*: Mensaje que señala que algunos de los índices de vértices proporcionados están fuera del rango permitido para el objeto.
- *Resultado exitoso*: Muestra la ruta más corta que conecta los vértices i1 e i2 en el objeto nombre\_objeto, indicando los vértices intermedios, la longitud de la ruta y su secuencia.

#### **Condiciones:**

- El objeto nombre\_objeto debe estar cargado en memoria para calcular la ruta.

#### **Descripción:**

Este comando encuentra la ruta más corta entre dos vértices específicos i1 e i2 en el objeto nombre\_objeto. Se calcula utilizando distancias euclidianas para definir un camino óptimo a través de los vértices intermedios, mostrando la secuencia de vértices en la ruta y la distancia total.

#### **- Comando: ruta\_corta\_centro i1 nombre\_objeto**

##### **Entradas:**

- i1: Índice del vértice desde el cual se buscará la ruta más corta hacia el centro.
- nombre\_objeto: Nombre del objeto 3D en el que se calculará la ruta.

##### **Salidas:**

- *Objeto no existe*: Mensaje que indica que el objeto nombre\_objeto no ha sido cargado en memoria.
- *Índice no existe*: Mensaje que señala que el índice del vértice proporcionado está fuera del rango permitido para el objeto.
- *Resultado exitoso*: Muestra la ruta más corta que conecta el vértice i1 con el centroide del objeto nombre\_objeto, indicando el centroide, los vértices intermedios y la distancia total de la ruta.

#### **Condiciones:**

- El objeto nombre\_objeto debe estar cargado en memoria para calcular la ruta.

#### **Descripción:**

Este comando identifica la ruta más corta desde un vértice i1 hasta el centro del objeto, calculado como el centroide de todos los vértices del objeto. El centroide se conecta con el vértice más cercano, y a partir de ahí se traza la ruta más corta utilizando distancias euclidianas.

## TADs

### - TAD Vértices

#### **Datos Mínimos:**

- x: Número entero, ubicación del punto en el eje x.
- y: Número entero, ubicación del punto en el eje y.
- z: Número entero, ubicación del punto en el eje z.

#### **Operaciones:**

- obtener\_x(): Obtener el valor de x en el vértice.
- obtener\_y(): Obtener el valor de y en el vértice.
- obtener\_z(): Obtener el valor de z en el vértice.
- set\_x(a): Fijar el valor de x en el vértice al valor de a.
- set\_y(a): Fijar el valor de y en el vértice al valor de a.
- set\_z(a): Fijar el valor de z en el vértice al valor de a.

### - TAD Caras

#### **Datos Mínimos:**

- vértices: Lista de vértices que forman una cara.

#### **Operaciones:**

- obtener\_vertices(): Obtener la lista de objetos vértices contenidos en la cara.

### - TAD Objeto 3D

#### **Datos Mínimos:**

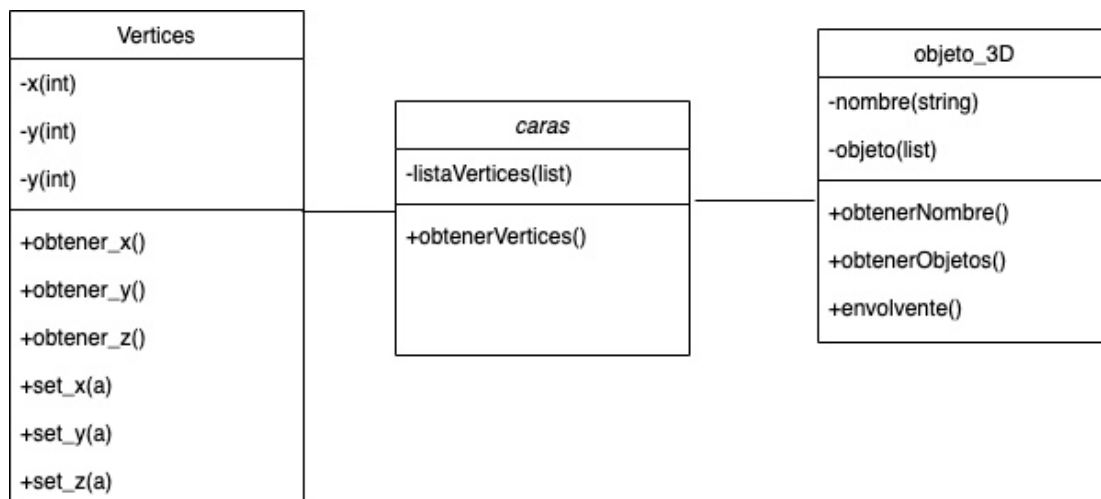
- Objeto: Lista de caras que componen un objeto 3D.

#### **Operaciones:**

- obtener\_objetos(): Obtener la lista de objetos 3D cargados en memoria.
- envolventeObjeto(): Buscar las coordenadas mínimas x, y, z para contener todos los objetos en su totalidad.

## ESQUEMA TADS

En el siguiente esquema se ve como los TADS se relacionan entre sí en este componente.



## Plan de Pruebas Comando ruta\_corta

### Objetivos:

El objetivo del plan de pruebas para el comando **ruta\_corta** fue garantizar su correcto funcionamiento mediante la validación exhaustiva de diferentes escenarios de uso, incluyendo casos exitosos y posibles errores. Este plan buscó verificar que el comando identificara de manera precisa la ruta más corta entre dos vértices de un objeto 3D, considerando tanto la correcta interpretación de la información de entrada como la precisión de los cálculos de distancia euclidiana. Adicionalmente, se evaluaron casos límite, como índices fuera de rango, vértices iguales y objetos inexistentes, asegurando que el sistema respondiera con mensajes claros y coherentes en cada situación, cumpliendo así con los requisitos de funcionalidad y robustez del proyecto.

Plan de pruebas: Función ruta_corta			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
Caso que busca la ruta	i1: 0	La ruta más corta que	La ruta más corta que

de 2 vértices conectados en un objeto existente	i2: 2 nombre_objeto: mesh1	conecta los vértices 0 y 2 del objeto nombre_objeto pasa por: 0 2 ; con una longitud de 57.1489.	conecta los vértices 0 y 2 del objeto nombre_objeto pasa por: 0 2 ; con una longitud de 57.1489.
Caso en el cual el objeto cuenta con los índices de los vértices inexistentes	i1:900 i2:3240 nombre_objeto: mesh1	Algunos de los índices de vertices estan fuera de rango para el objeto mesh1	Algunos de los índices de vertices estan fuera de rango para el objeto mesh1
Caso en el que el objeto tiene índice de ambos vértices es igual	i1:1 i2:1 nombre_objeto:mesh1	Los índices de los vertices dados son iguales.	Los índices de los vertices dados son iguales.

Se implementó un plan de pruebas para evaluar el correcto funcionamiento de la función **ruta\_corta**, con tres casos distintos que cubrían diferentes escenarios posibles dentro del proyecto. Al ejecutar las pruebas, los resultados obtenidos coincidieron plenamente con los esperados, demostrando que la función opera correctamente en cada uno de los casos analizados. Esto confirma que el plan de pruebas fue un éxito, validando tanto la implementación de la función como su capacidad para manejar adecuadamente las situaciones previstas.

## Planes de Prueba en la terminal

### Plan de Prueba 1

```
ProyectoEntregaFinal — EntregaFinalEst — 80x24
Introduce el nombre del archivo para cargar el objeto$ Prueba2.txt
Objeto mesh1 cargado exitosamente.
Objeto mesh2 cargado exitosamente.

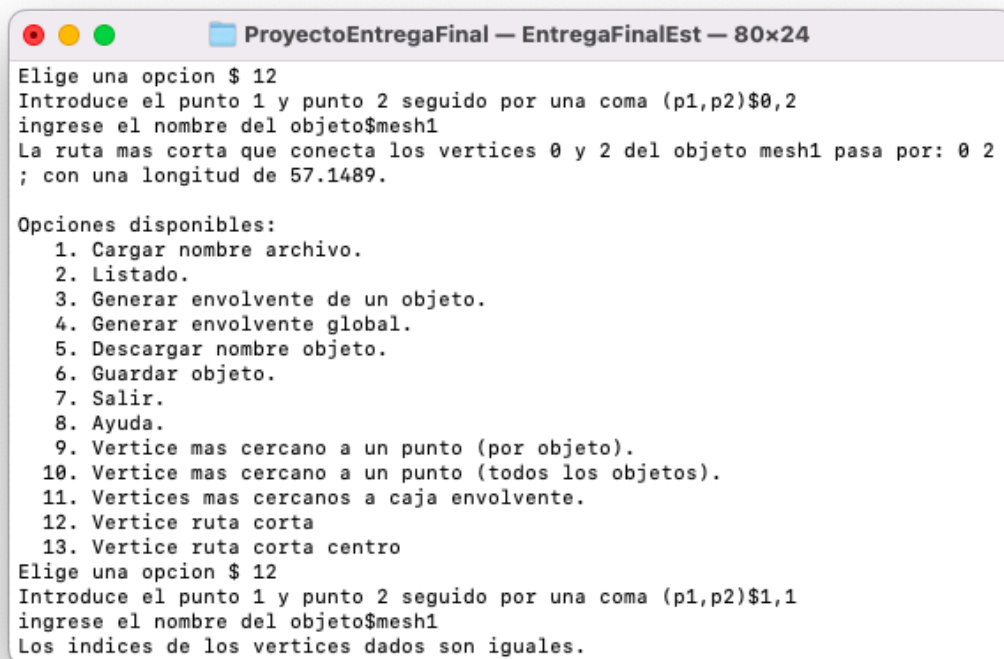
Opciones disponibles:
  1. Cargar nombre archivo.
  2. Listado.
  3. Generar envoltente de un objeto.
  4. Generar envoltente global.
  5. Descargar nombre objeto.
  6. Guardar objeto.
  7. Salir.
  8. Ayuda.
  9. Vertice mas cercano a un punto (por objeto).
 10. Vertice mas cercano a un punto (todos los objetos).
 11. Vertices mas cercanos a caja envoltente.
 12. Vertice ruta corta
 13. Vertice ruta corta centro
Elige una opcion $ 12
Introduce el punto 1 y punto 2 seguido por una coma (p1,p2)$0,2
ingrese el nombre del objeto$mesh1
La ruta mas corta que conecta los vertices 0 y 2 del objeto mesh1 pasa por: 0 2
; con una longitud de 57.1489.
```

## Plan de Prueba 2

```
ProyectoEntregaFinal — EntregaFinalEst — 80x24
Elige una opcion $ 12
Introduce el punto 1 y punto 2 seguido por una coma (p1,p2)$1,1
ingrese el nombre del objeto$mesh1
Los indices de los vertices dados son iguales.

Opciones disponibles:
  1. Cargar nombre archivo.
  2. Listado.
  3. Generar envoltente de un objeto.
  4. Generar envoltente global.
  5. Descargar nombre objeto.
  6. Guardar objeto.
  7. Salir.
  8. Ayuda.
  9. Vertice mas cercano a un punto (por objeto).
 10. Vertice mas cercano a un punto (todos los objetos).
 11. Vertices mas cercanos a caja envoltente.
 12. Vertice ruta corta
 13. Vertice ruta corta centro
Elige una opcion $ 12
Introduce el punto 1 y punto 2 seguido por una coma (p1,p2)$900,3240
ingrese el nombre del objeto$mesh1
Algunos de los indices de vertices estan fuera de rango para el objeto mesh1.
```

## Plan de Prueba 3



```
ProyectoEntregaFinal — EntregaFinalEst — 80x24
Elige una opcion $ 12
Introduce el punto 1 y punto 2 seguido por una coma (p1,p2)$0,2
ingrese el nombre del objeto$mesh1
La ruta mas corta que conecta los vertices 0 y 2 del objeto mesh1 pasa por: 0 2
; con una longitud de 57.1489.

Opciones disponibles:
1. Cargar nombre archivo.
2. Listado.
3. Generar envoltente de un objeto.
4. Generar envoltente global.
5. Descargar nombre objeto.
6. Guardar objeto.
7. Salir.
8. Ayuda.
9. Vertice mas cercano a un punto (por objeto).
10. Vertice mas cercano a un punto (todos los objetos).
11. Vertices mas cercanos a caja envoltente.
12. Vertice ruta corta
13. Vertice ruta corta centro
Elige una opcion $ 12
Introduce el punto 1 y punto 2 seguido por una coma (p1,p2)$1,1
ingrese el nombre del objeto$mesh1
Los indices de los vertices dados son iguales.
```

## Resultados Obtenidos

Durante el desarrollo del proyecto, logramos implementar un sistema interactivo funcional que permite gestionar y manipular eficientemente mallas poligonales en 3D. Los comandos diseñados para cargar, listar, calcular cajas envoltentes y encontrar vértices más cercanos, entre otras funcionalidades, operaron correctamente, cumpliendo con los requisitos planteados. Además, se validaron casos de error, garantizando que el sistema respondiera con mensajes claros y precisos para cada situación. Este enfoque estructurado no solo permitió mantener la integridad de los datos en memoria, sino también optimizó los procesos de consulta y manipulación de los objetos 3D.

En cuanto al cálculo de rutas más cortas, los algoritmos implementados demostraron ser precisos y eficientes, ofreciendo resultados confiables incluso para modelos complejos. Por ejemplo, el sistema fue capaz de identificar rutas óptimas entre vértices específicos y calcular distancias hacia puntos centrales en diversos objetos, lo que evidencia la robustez de la solución. La integración de todos los componentes dentro de una interfaz de consola amigable brindó una experiencia fluida, asegurando que el sistema fuera accesible y útil tanto para propósitos educativos como prácticos. En general, el proyecto no solo cumplió con los objetivos, sino que también sirvió como un ejercicio valioso para consolidar habilidades en programación, manejo de estructuras de datos y algoritmos geométricos.