

INFORMÁTICA GRÁFICA

DEPARTAMENTO DE LENGUAJES
Y SISTEMAS INFORMÁTICOS

PRÁCTICA 3: MODELOS JERÁRQUICOS

JAIME MARTÍNEZ BRAVO

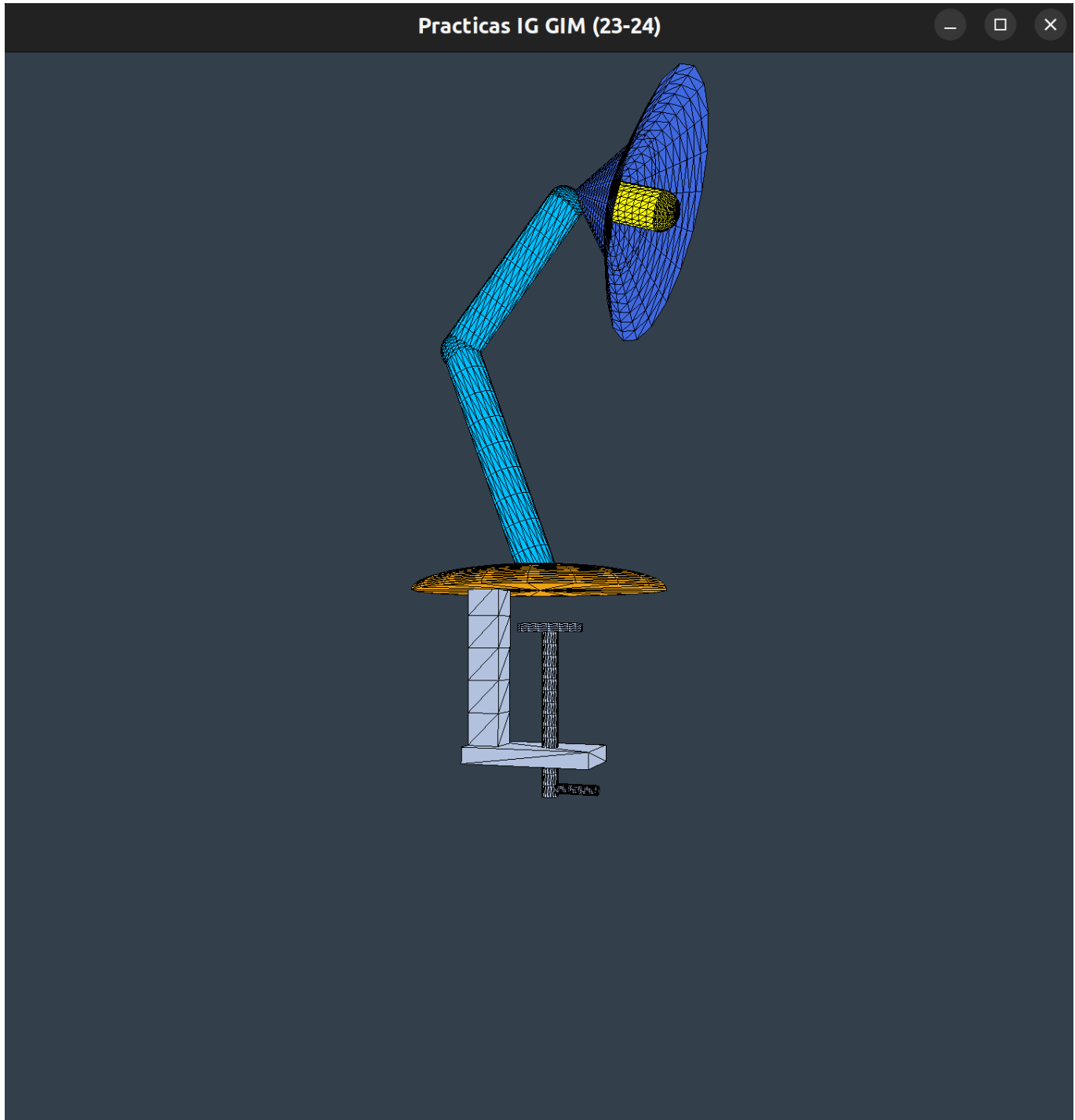
DOBLE GRADO EN
INGENIERÍA INFORMÁTICA
Y MATEMÁTICAS

CURSO: 2023-24

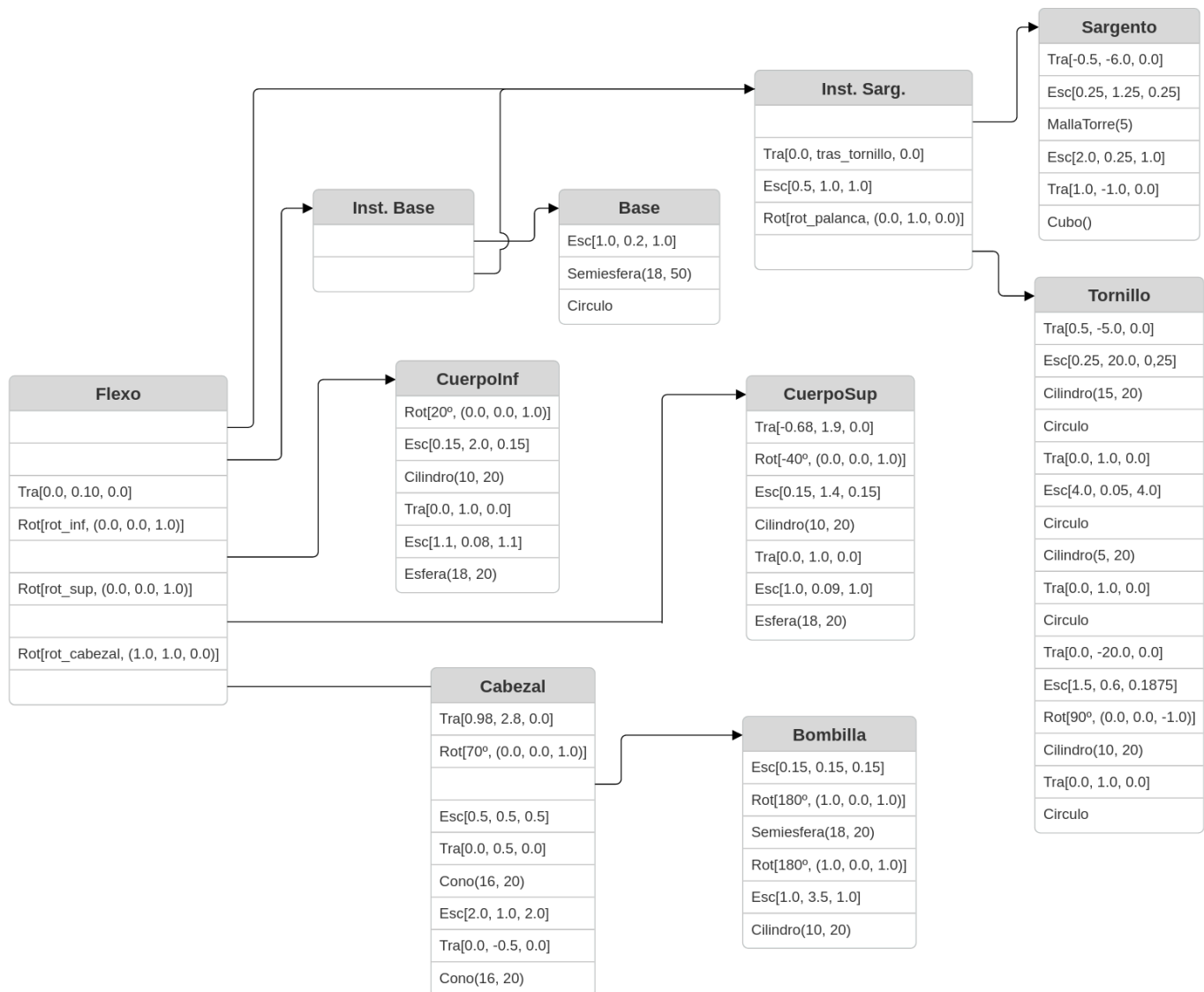
Índice

1. Modelo	3
2. Grafo de escena	4
3. Nodos del grafo	5
3.1. Clases derivadas de <code>NodoGrafoEscena</code>	5
3.2. Clases de objetos simples	7
4. Parámetros de libertad	9

1. Modelo



2. Grafo de escena



3. Nodos del grafo

3.1. Clases derivadas de NodoGrafoEscena

- Flexo (nodo raíz)
 - Parámetros de libertad: 0, 1, 2
 - Color: varios
 - Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 32-59
- Base
 - Parámetros de libertad: no tiene
 - Color: (0.93, 0.64, 0.09)
 - Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 94-101
- Sargento
 - Parámetros de libertad: 3, 4
 - Color: (0.7, 0.76, 0.87)
 - Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 103-114
- Tornillo
 - Parámetros de libertad: no tiene
 - Color: no tiene
 - Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 116-137

■ **CuerpoInf**

- Parámetros de libertad: no tiene
- Color: (0.0, 0.75, 1.0)
- Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 139-150

■ **CuerpoSup**

- Parámetros de libertad: no tiene
- Color: (0.0, 0.75, 1.0)
- Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 152-165

■ **Cabecal**

- Parámetros de libertad: no tiene
- Color: (0.25, 0.41, 0.88)
- Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 167-183

■ **Bombilla**

- Parámetros de libertad: no tiene
- Color: (1.0, 1.0, 0.0)
- Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 185-196

3.2. Clases de objetos simples

■ Circulo

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `modelo-jer.h`, `mall-ind.cpp`
 - Implementación del constructor: líneas 200-210

■ Semiesfera

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `modelo-jer.h`, `modelo-jer.cpp`
 - Implementación del constructor: líneas 212-225

■ Cubo

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `mall-ind.h`, `mall-ind.cpp`
 - Implementación del constructor: líneas 284-312

■ MallaTorre

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `mall-ind.h`, `mall-ind.cpp`
 - Implementación del constructor: líneas 560-584

■ Cilindro

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `malla-revol.h`, `malla-revol.cpp`
 - Implementación del constructor: líneas 106-118

■ Cono

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `malla-revol.h`, `malla-revol.cpp`
 - Implementación del constructor: líneas 123-135

■ Esfera

- Parámetros de libertad: no tiene
- Color: no tiene
- Archivos de declaración/implementación: `malla-revol.h`, `malla-revol.cpp`
 - Implementación del constructor: líneas 140-153

4. Parámetros de libertad

- Ángulo de rotación del cuerpo inferior del flexo
 - Identificador: 0
 - Nodo: Flexo
 - Matriz dependiente: `pm_rot_inf`
 - Descripción: es una rotación oscilante en torno al eje Z, con una frecuencia de 0.3 oscilaciones por segundo, y una amplitud de $\pi/7$ radianes.
 - Expresión:

```
*pm_rot_inf = rotate(float(-M_PI/7)+float(M_PI/7)*sin(float(M_PI*0.3*t_sec)),  
vec3 0.0,0.0,1.0 );
```
- Ángulo de rotación del cuerpo superior del flexo
 - Identificador: 1
 - Nodo: Flexo
 - Matriz dependiente: `pm_rot_sup`
 - Descripción: es una rotación oscilante en torno al centro de la esfera del cuerpo inferior, con una frecuencia de 0.3 oscilaciones por segundo, y una amplitud de $\pi/3$ radianes.
 - Expresión:

```
*pm_rot_sup = (translate(vec3(-0.68, 1.9, 0.0))) *  
(rotate(float(M_PI/3)+float(M_PI/3)*sin(float(M_PI*0.3*t_sec)),  
vec3 0.0,0.0,1.0 )) * (translate(vec3(0.68, -1.9, 0.0)));
```
- Ángulo de rotación del cabezal del flexo
 - Identificador: 2
 - Nodo: Flexo
 - Matriz dependiente: `pm_rot_cabezal`
 - Descripción: es una rotación en torno al punto de unión del cuerpo inferior y el superior, con una frecuencia de 0.3 oscilaciones por segundo.

- Expresión:


```
*pm_rot_cabecal = (translate(vec3(-1.1,1.7,0.0))) *
(rotate(float(2*M_PI*0.2)*t_sec, vec3 1.0,1.0,0.0 )) *
(translate(vec3(1.1,-1.7,0.0)));
```
- Distancia de desplazamiento del tornillo del sargento
 - Identificador: 3
 - Nodo: Sargento
 - Matriz dependiente: `pm_tras_tornillo`
 - Descripción: es un desplazamiento oscilante vertical, con una frecuencia de 0.3 oscilaciones por segundo, y una amplitud de 4 unidades de distancia.
 - Expresión:


```
*pm_tras_tornillo = translate(vec3(0.0, -4-4*sin(2*M_PI*0.15*t_sec), 0.0));
```
- Ángulo de rotación del tornillo del sargento
 - Identificador: 4
 - Nodo: Sargento
 - Matriz dependiente: `pm_rot_tornillo`
 - Descripción: es una rotación oscilante en torno al eje vertical del tornillo, con una frecuencia de 0.3 oscilaciones por segundo, y una amplitud de $\pi/3$ radianes.
 - Expresión:


```
*pm_rot_tornillo = (translate(vec3(0.5, 4.5, 0.0))) *
(rotate(float(2.5*M_PI)+float(2.5*M_PI)*sin(float(M_PI*0.3*t_sec)),
vec3 0.0,1.0,0.0 )) * (translate(vec3(-0.5, -4.5, 0.0)));
```