UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
Facultad de Ingeniería de Producción y Servicios
Escuela Profesional de Ciencia de la Computación

TRANSFORMACIONES

PRÁCTICA N°04

ALUMNO: YORDY W. SANTOS APAZA

DOCENTE: M.Sc. VICENTE E. MACHACA ARCEDA

Fecha de entrega: 5 de mayo de 2021

Arequipa, Perú

Índice de Contenidos

Índice de Figuras

Índice de Códigos

1. Ejercicio #1:

Cree una matriz de transformación para trasladar un objeto 0.01 unidades en el eje x.

Código 1: Código_01

```
const transformation = new THREE.Matrix4();
transformation.set(
    1, 0, 0, 0.01,
    0, 1, 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
 controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
     controls.update()
    renderer.render( scene, camera)
}
animate()
```

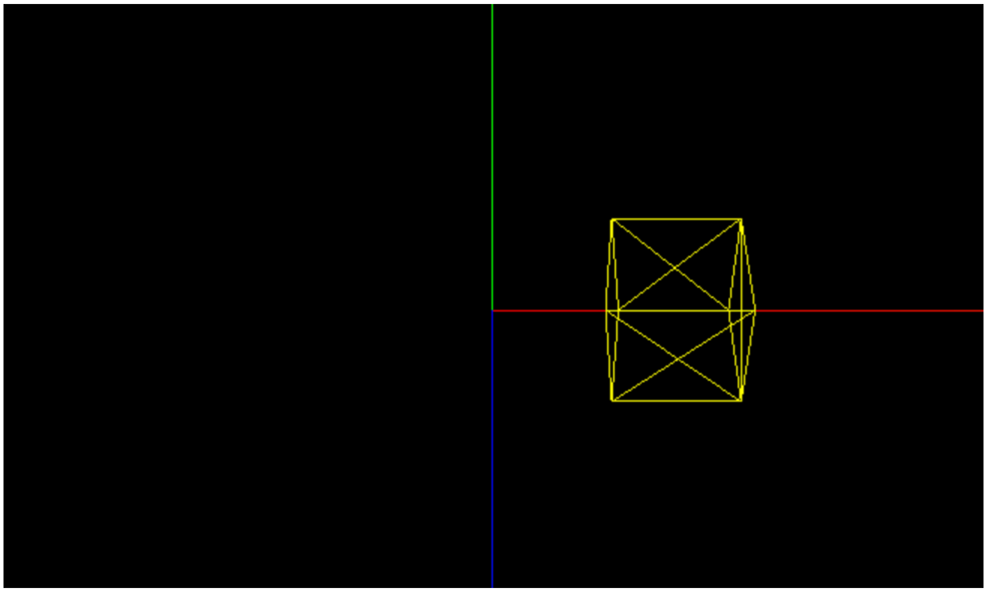

Figura 1: Gráfico_01

2. Ejercicio #2:

Cree una matriz de transformación para trasladar un objeto 0.01 unidades en el eje x y 0.02 unidades en el eje z.

Código 2: Código_02

```
const transformation = new THREE.Matrix4();
transformation.set(
    1, 0, 0, 0.01,
    0, 1, 0, 0,
    0, 0, 1, 0.02,
    0, 0, 0, 1
);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
```

```
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
 controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
     controls.update()
    renderer.render( scene, camera)
}
animate()
```

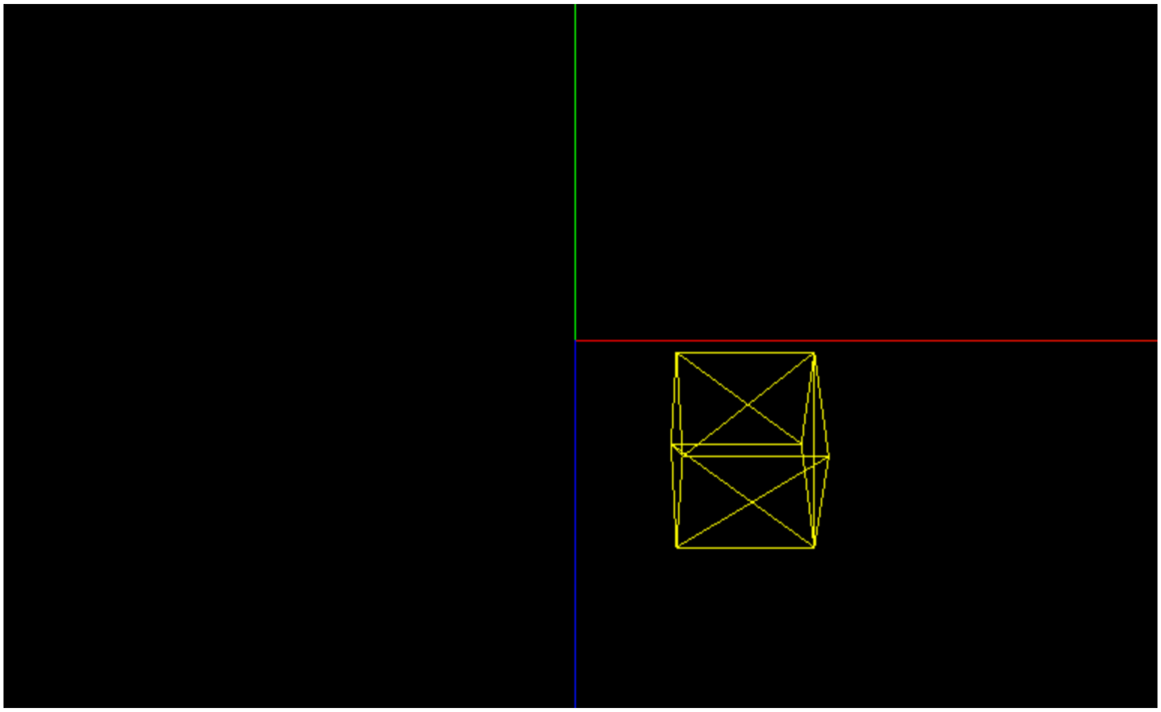

Figura 2: Gráfico_02

3. Ejercicio #3:

Cree una matriz de transformación que escale un objeto 1.001 unidades en el eje y.

Código 3: Código_03

```
const transformation = new THREE.Matrix4();
transformation.set(
    1, 0, 0, 0,
    0, 1.001, 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
 controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
     controls.update()
    renderer.render( scene, camera)
}
animate()
```

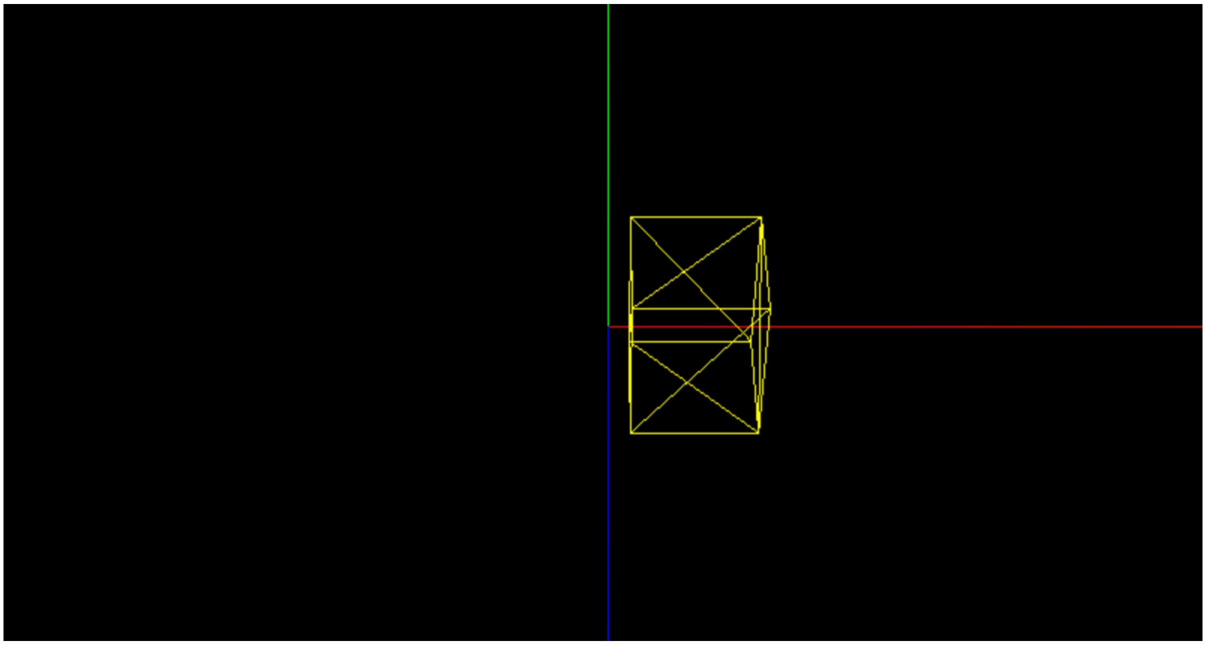

Figura 3: Gráfico_03

4. Ejercicio #4:

Cree una matriz de transformación que escale un objeto 1.001 unidades en todos los ejes.

Código 4: Código_04

```
const transformation = new THREE.Matrix4();
transformation.set(
    1.001, 0, 0, 0,
    0, 1.001, 0, 0,
    0, 0, 1.001, 0,
    0, 0, 0, 1
);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
    controls.update()
    renderer.render( scene, camera)
}
animate()
```

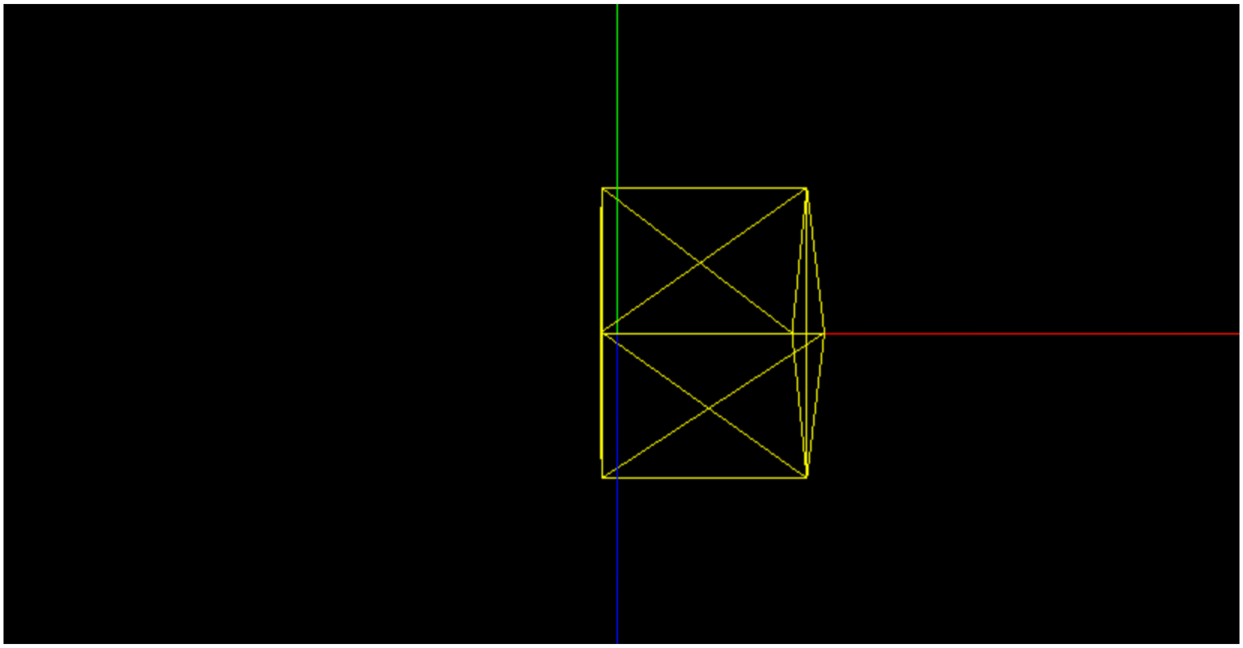

Figura 4: Gráfico_04

5. Ejercicio #5:

Cree una matriz de transformación que gire un objeto 0.01 unidades en el eje z.

Código 5: Código_05

```
const transformation = new THREE.Matrix4();
transformation.set(
    Math.cos(0.01), -Math.sin(0.01), 0, 0,
    Math.sin(0.01), Math.cos(0.01), 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
```

```
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
    controls.update()
    renderer.render( scene, camera)
}
animate()
```

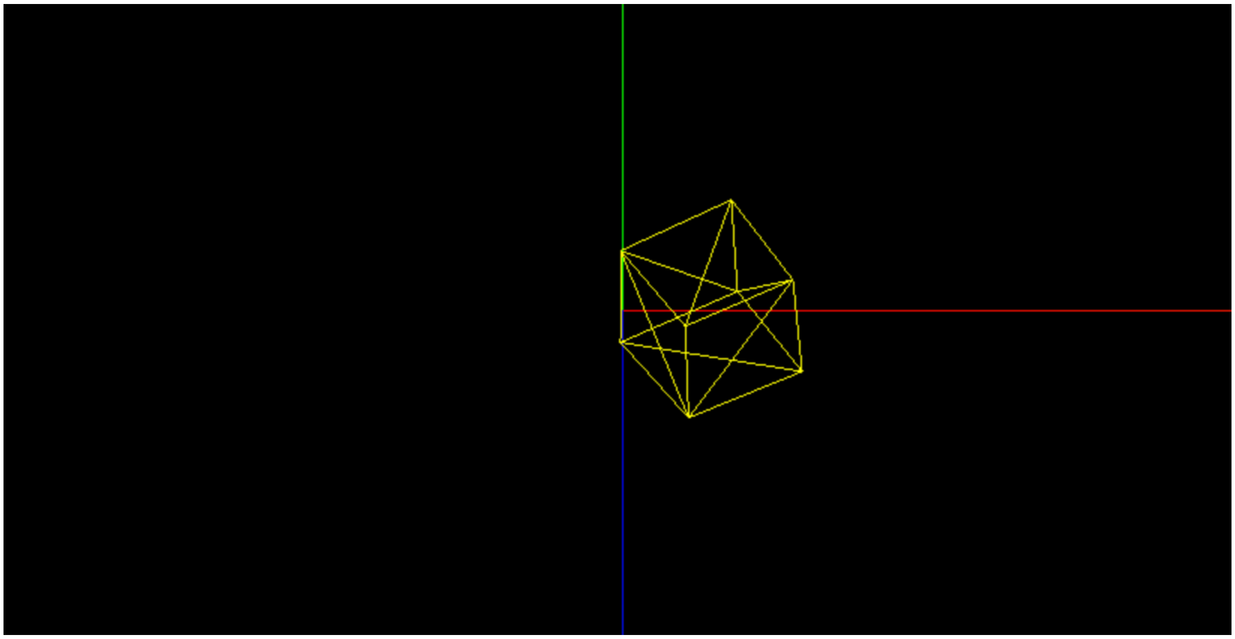

Figura 5: Gráfico_05

6. Ejercicio #6:

Cree una matriz de transformación que gire un objeto 0.01 unidades en el eje z y 0.02 unidades en el eje x. Aquí, deberá multiplicar matrices.

Código 6: Código_06

```
const rotaZ = new THREE.Matrix4();
rotaZ.set(
    Math.cos(0.01), -Math.sin(0.01), 0, 0,
    Math.sin(0.01), Math.cos(0.01), 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
const rotaX = new THREE.Matrix4();
```

```
rotaX.set(
    1, 0, 0, 0,
    0, Math.cos(0.02), -Math.sin(0.02), 0,
    0, Math.sin(0.02), Math.cos(0.02), 0,
    0, 0, 0, 1
);
const transformation = rotaZ.multiply(rotaX);
var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true });
var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
cube.position.x = 2
cube.position.y = 2
cube.position.z = 2
var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
camera.position.z = 30;
camera.position.y = 30;
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x000000)
scene.add(cube);
axesHelper = new THREE.AxesHelper( 100 );
scene.add( axesHelper );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
 controls.enableDamping = true;
var animate = () => {
    requestAnimationFrame(animate)
    cube.geometry.applyMatrix4( transformation )
     controls.update()
    renderer.render( scene, camera)
}
animate()
```

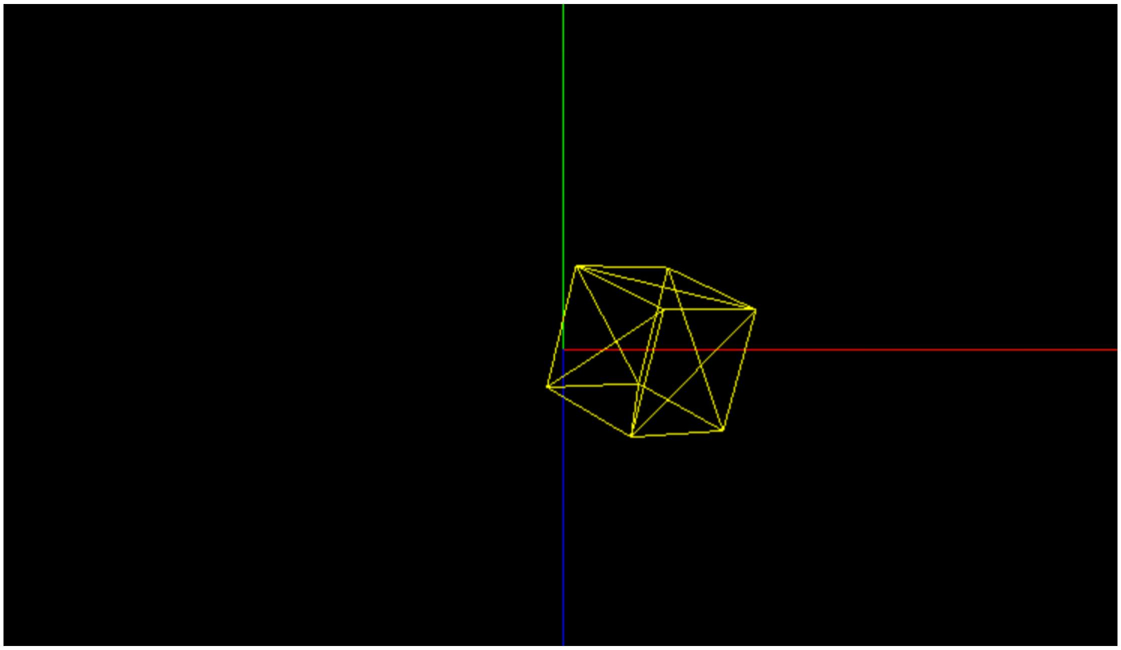

Figura 6: Gráfico_06

7. Ejercicio #7:

Cree una matriz de transformación que traslade un objeto 0.02 unidades en el eje x, que escale 1.002 unidades en todos los ejes y que gire 0.01 unidades en el eje x y y. Aquí, deberá multiplicar matrices.

Código 7: Código_07

```
var geometry = new THREE.BoxGeometry(3, 3, 3, 3, 3, 3);
var material = new THREE.MeshBasicMaterial({color: 0xffff00, wireframe: true});
var cube = new THREE.Mesh(geometry, material);
const translation  = new THREE.Matrix4();
 translation . set (
    1,  0,  0,  0.02,
    0,  1,  0,  0,
    0,  0,  1,  0,
    0,  0,  0,  1
);
const scala  = new THREE.Matrix4();
scala . set (
    1.002,  0,  0,  0,
    0,  1.002,  0,  0,
    0,  0,  1.002,  0,
    0,  0,  0,  1
);
const rotaX = new THREE.Matrix4();
rotaX.set(
    1,  0,  0,  0,
    0, Math.cos(0.01), -Math.sin(0.01), 0,
    0, Math.sin(0.01), Math.cos(0.01), 0,
    0, 0, 0, 1
);
const rotaY = new THREE.Matrix4();
rotaY.set(
    Math.cos(0.01), 0, -Math.sin(0.01), 0,
    0, 1, 0, 0,
    Math.sin(0.01), 0, Math.cos(0.01), 0,
    0, 0, 0, 1
);
var ejer07 = translation.multiply(scala).multiply(rotaY).multiply(rotaX);
var scene = new THREE.Scene();
const axesHelper = new THREE.AxesHelper(5);
scene.add(axesHelper);
scene.add(cube);
var camera = new THREE.PerspectiveCamera(30, window.innerWidth/window.innerHeight, 0.1, 1000);
camera.position.set(20, 20, 20);
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
var controls = new THREE.OrbitControls( camera, renderer.domElement );
controls.minDistance = 5;
controls.maxDistance = 100;
var animate = function () {
    requestAnimationFrame(animate);
    cube.geometry.applyMatrix4(ejer07);
    renderer.render(scene, camera);
}
animate();
```

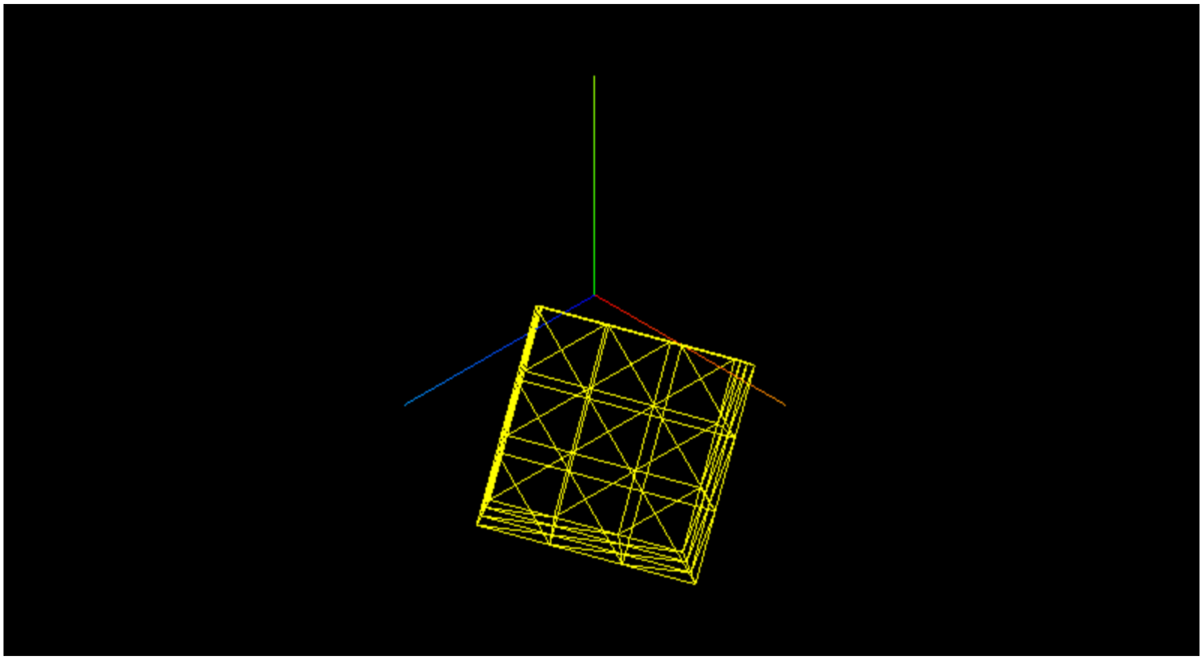

Figura 7: Gráfico_07