

# 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

1. Ejercicio #1:	1
2. Ejercicio #2:	2
3. Ejercicio #3:	3
4. Ejercicio #4:	5
5. Ejercicio #5:	6
6. Ejercicio #6:	7
7. Ejercicio #7:	9

## Índice de Figuras

1. Gráfico_01 . . . . .	2
2. Gráfico_02 . . . . .	3
3. Gráfico_03 . . . . .	4
4. Gráfico_04 . . . . .	6
5. Gráfico_05 . . . . .	7
6. Gráfico_06 . . . . .	9
7. Gráfico_07 . . . . .	11

## Índice de Códigos

1. Código_01 . . . . .	1
2. Código_02 . . . . .	2
3. Código_03 . . . . .	3
4. Código_04 . . . . .	5
5. Código_05 . . . . .	6
6. Código_06 . . . . .	7
7. Código_07 . . . . .	9

## 1. Ejercicio #1:

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

Código 1: Código\_01

```
1 const transformation = new THREE.Matrix4();
2 transformation.set(
3     1, 0, 0, 0.01,
4     0, 1, 0, 0,
5     0, 0, 1, 0,
6     0, 0, 0, 1
7 );
8 var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
9 var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true } );
10 var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
11 cube.position.x = 2
12 cube.position.y = 2
13 cube.position.z = 2
14 var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
15 camera.position.z = 30;
16 camera.position.y = 30;
17 var scene = new THREE.Scene();
18 scene.background = new THREE.Color(0x000000)
19 scene.add(cube);
20 axesHelper = new THREE.AxesHelper( 100 );
21 scene.add( axesHelper );
22 var renderer = new THREE.WebGLRenderer();
23 renderer.setSize( window.innerWidth, window.innerHeight );
24 document.body.appendChild( renderer.domElement );
25 var controls = new THREE.OrbitControls( camera, renderer.domElement );
26 controls.enableDamping = true;
27 var animate = () => {
28     requestAnimationFrame(animate)
29     cube.geometry.applyMatrix4( transformation )
30     controls.update()
31     renderer.render( scene, camera)
32 }
33 animate()
```

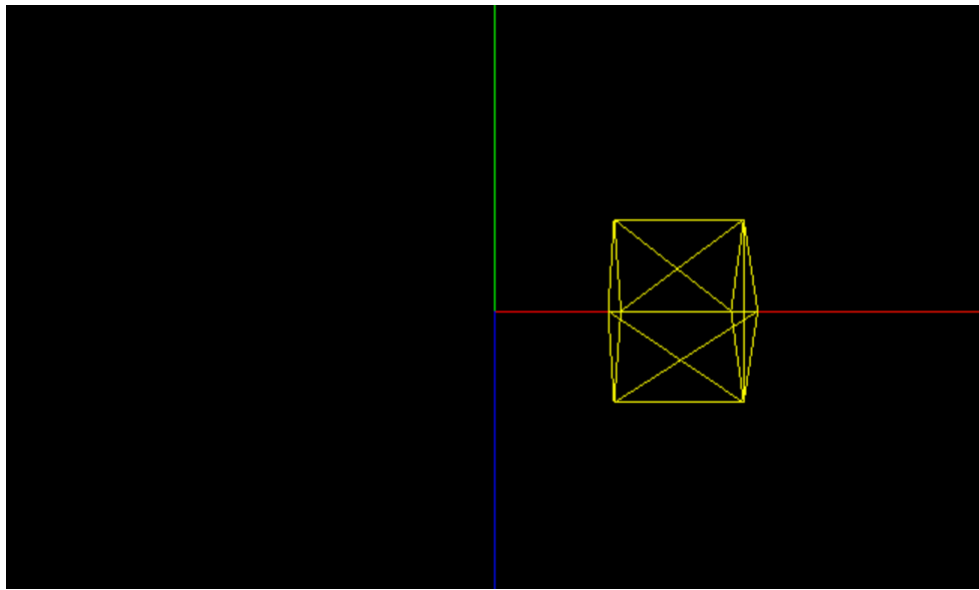


Figura 1: Gráfico\_01

## 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

```
1 const transformation = new THREE.Matrix4();
2 transformation.set(
3     1, 0, 0, 0.01,
4     0, 1, 0, 0,
5     0, 0, 1, 0.02,
6     0, 0, 0, 1
7 );
8 var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
9 var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true } );
10 var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
11 cube.position.x = 2
12 cube.position.y = 2
13 cube.position.z = 2
14 var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
15 camera.position.z = 30;
16 camera.position.y = 30;
17 var scene = new THREE.Scene();
18 scene.background = new THREE.Color(0x000000)
19 scene.add(cube);
20 axesHelper = new THREE.AxesHelper( 100 );
21 scene.add( axesHelper );
22 var renderer = new THREE.WebGLRenderer();
23 renderer.setSize( window.innerWidth, window.innerHeight );
```

```
24 document.body.appendChild( renderer.domElement );
25 var controls = new THREE.OrbitControls( camera, renderer.domElement );
26 controls.enableDamping = true;
27 var animate = () => {
28     requestAnimationFrame(animate)
29     cube.geometry.applyMatrix4( transformation )
30     controls.update()
31     renderer.render( scene, camera)
32 }
33 animate()
```

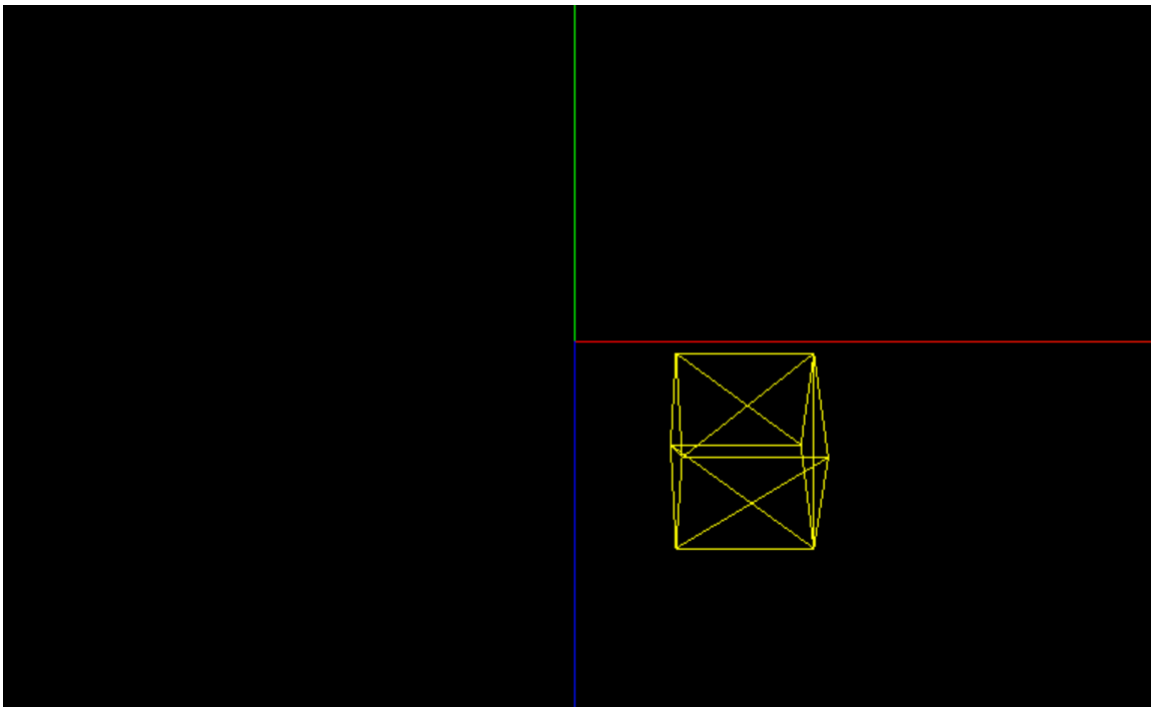


Figura 2: Gráfico\_02

### 3. Ejercicio #3:

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

Código 3: Código\_03

```
1 const transformation = new THREE.Matrix4();
2 transformation.set(
3     1, 0, 0, 0,
4     0, 1.001, 0, 0,
5     0, 0, 1, 0,
6     0, 0, 0, 1
7 );
8 var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
```

```
9  var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true } );
10 var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
11 cube.position.x = 2
12 cube.position.y = 2
13 cube.position.z = 2
14 var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
15 camera.position.z = 30;
16 camera.position.y = 30;
17 var scene = new THREE.Scene();
18 scene.background = new THREE.Color(0x000000)
19 scene.add(cube);
20 axesHelper = new THREE.AxesHelper( 100 );
21 scene.add( axesHelper );
22 var renderer = new THREE.WebGLRenderer();
23 renderer.setSize( window.innerWidth, window.innerHeight );
24 document.body.appendChild( renderer.domElement );
25 var controls = new THREE.OrbitControls( camera, renderer.domElement );
26 controls.enableDamping = true;
27 var animate = () => {
28     requestAnimationFrame(animate)
29     cube.geometry.applyMatrix4( transformation )
30     controls.update()
31     renderer.render( scene, camera)
32 }
33 animate()
```

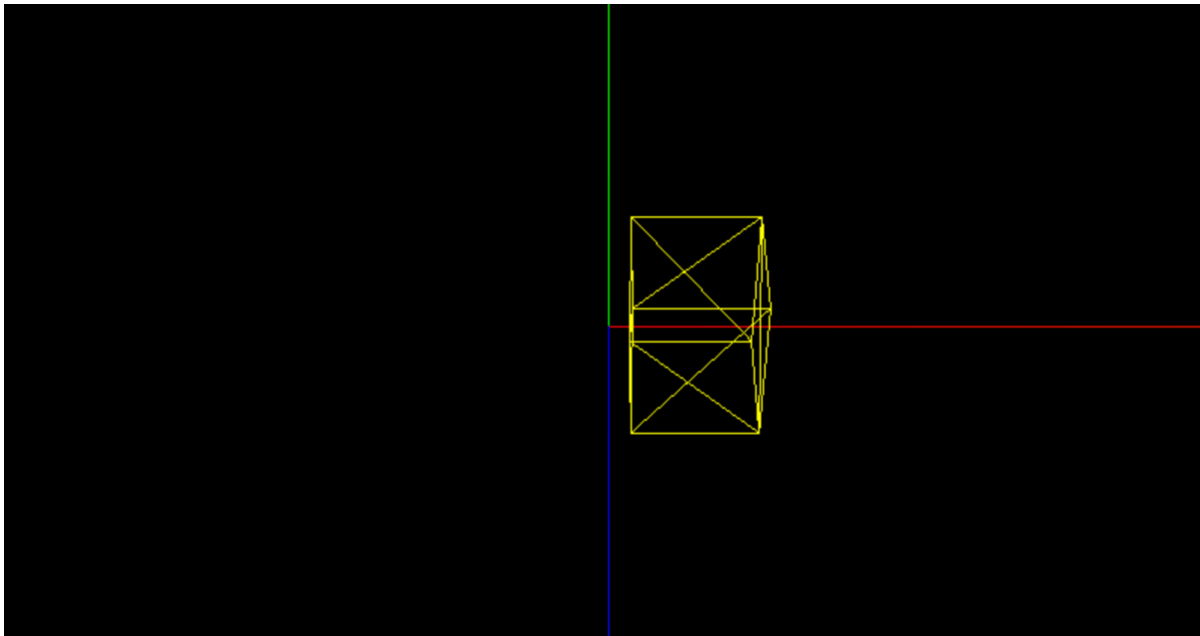


Figura 3: Gráfico\_03

## 4. Ejercicio #4:

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

Código 4: Código\_04

```
1 const transformation = new THREE.Matrix4();
2 transformation.set(
3     1.001, 0, 0, 0,
4     0, 1.001, 0, 0,
5     0, 0, 1.001, 0,
6     0, 0, 0, 1
7 );
8 var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
9 var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true} );
10 var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
11 cube.position.x = 2
12 cube.position.y = 2
13 cube.position.z = 2
14 var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
15 camera.position.z = 30;
16 camera.position.y = 30;
17 var scene = new THREE.Scene();
18 scene.background = new THREE.Color(0x000000)
19 scene.add(cube);
20 axesHelper = new THREE.AxesHelper( 100 );
21 scene.add( axesHelper );
22 var renderer = new THREE.WebGLRenderer();
23 renderer.setSize( window.innerWidth, window.innerHeight );
24 document.body.appendChild( renderer.domElement );
25 var controls = new THREE.OrbitControls( camera, renderer.domElement );
26 controls.enableDamping = true;
27 var animate = () => {
28     requestAnimationFrame(animate)
29     cube.geometry.applyMatrix4( transformation )
30     controls.update()
31     renderer.render( scene, camera)
32 }
33 animate()
```

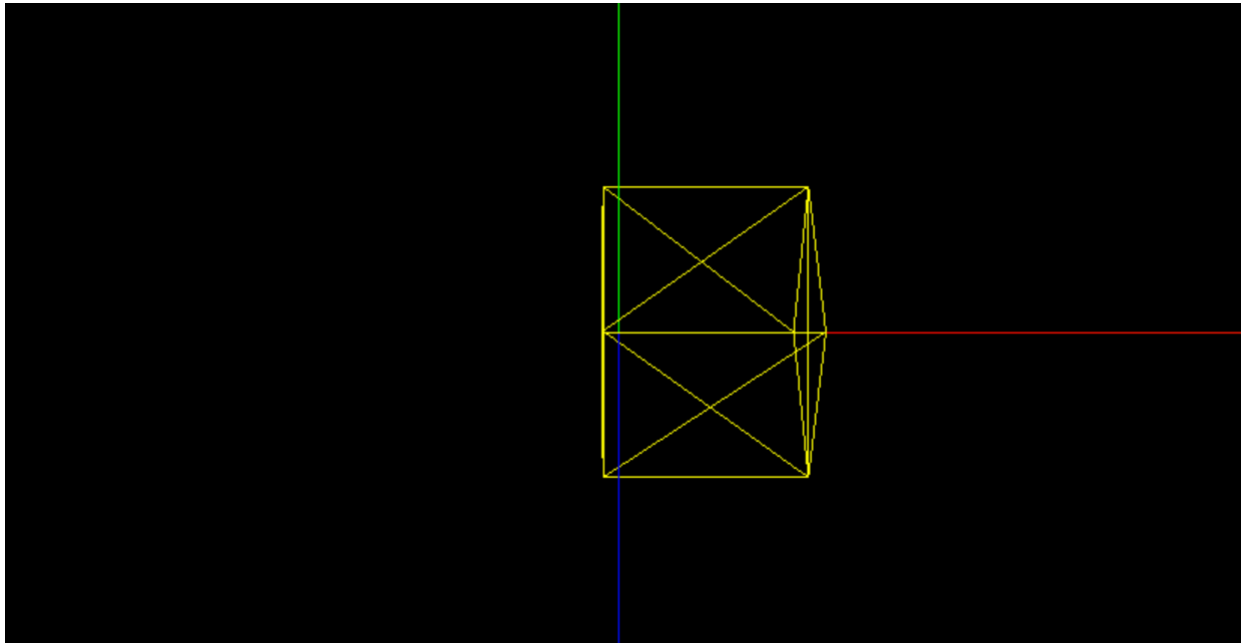


Figura 4: Gráfico\_04

## 5. Ejercicio #5:

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

Código 5: Código\_05

```
1 const transformation = new THREE.Matrix4();
2 transformation.set(
3     Math.cos(0.01), -Math.sin(0.01), 0, 0,
4     Math.sin(0.01), Math.cos(0.01), 0, 0,
5     0, 0, 1, 0,
6     0, 0, 0, 1
7 );
8 var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);
9 var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true} );
10 var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );
11 cube.position.x = 2
12 cube.position.y = 2
13 cube.position.z = 2
14 var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);
15 camera.position.z = 30;
16 camera.position.y = 30;
17 var scene = new THREE.Scene();
18 scene.background = new THREE.Color(0x000000)
19 scene.add(cube);
20 axesHelper = new THREE.AxesHelper( 100 );
21 scene.add( axesHelper );
22 var renderer = new THREE.WebGLRenderer();
```



```

23 renderer.setSize( window.innerWidth, window.innerHeight );
24 document.body.appendChild( renderer.domElement );
25 var controls = new THREE.OrbitControls( camera, renderer.domElement );
26 controls.enableDamping = true;
27 var animate = () => {
28     requestAnimationFrame(animate)
29     cube.geometry.applyMatrix4( transformation )
30     controls.update()
31     renderer.render( scene, camera)
32 }
33 animate()

```

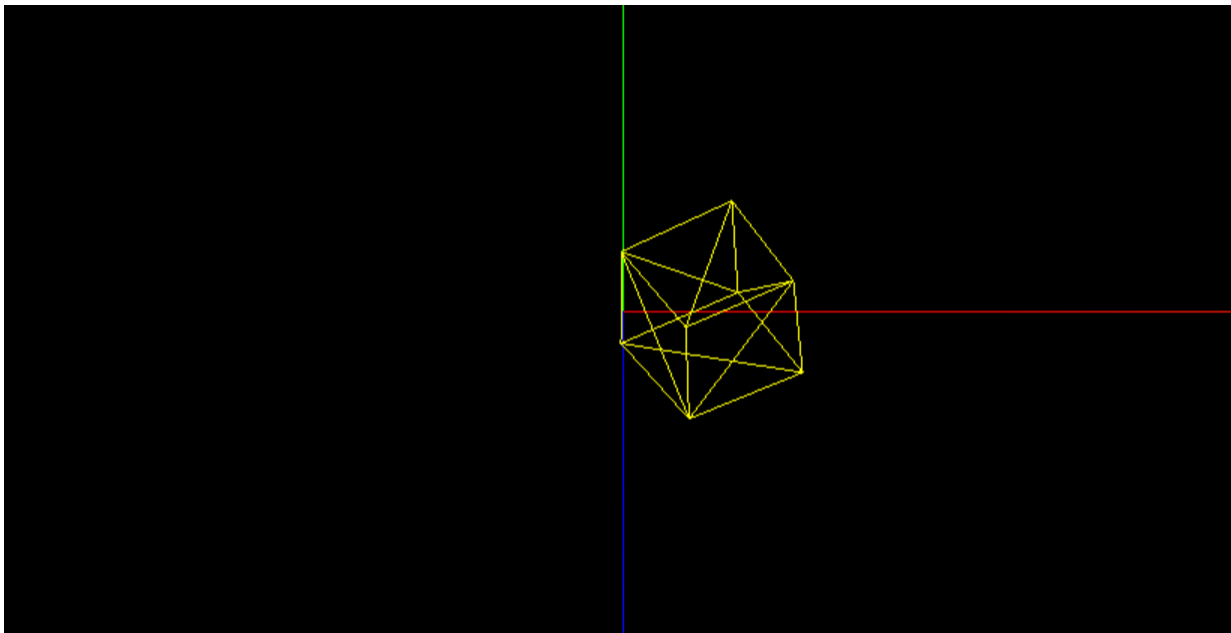


Figura 5: Gráfico\_05

## 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

```

1 const rotaZ = new THREE.Matrix4();
2 rotaZ.set(
3     Math.cos(0.01), -Math.sin(0.01), 0, 0,
4     Math.sin(0.01), Math.cos(0.01), 0, 0,
5     0, 0, 1, 0,
6     0, 0, 0, 1
7 );
8 const rotaX = new THREE.Matrix4();

```

```
9  rotaX.set(  
10    1, 0, 0, 0,  
11    0, Math.cos(0.02), -Math.sin(0.02), 0,  
12    0, Math.sin(0.02), Math.cos(0.02), 0,  
13    0, 0, 0, 1  
14  );  
15  const transformation = rotaZ.multiply(rotaX);  
16  var cubeGeometry = new THREE.BoxGeometry(3, 3, 3);  
17  var cubeMaterial = new THREE.MeshBasicMaterial( {color: 0xffff00, wireframe: true } );  
18  var cube = new THREE.Mesh( cubeGeometry, cubeMaterial );  
19  cube.position.x = 2  
20  cube.position.y = 2  
21  cube.position.z = 2  
22  var camera = new THREE.PerspectiveCamera(30,window.innerWidth/window.innerHeight);  
23  camera.position.z = 30;  
24  camera.position.y = 30;  
25  var scene = new THREE.Scene();  
26  scene.background = new THREE.Color(0x000000)  
27  scene.add(cube);  
28  axesHelper = new THREE.AxesHelper( 100 );  
29  scene.add( axesHelper );  
30  var renderer = new THREE.WebGLRenderer();  
31  renderer.setSize( window.innerWidth, window.innerHeight );  
32  document.body.appendChild( renderer.domElement );  
33  var controls = new THREE.OrbitControls( camera, renderer.domElement );  
34  controls.enableDamping = true;  
35  var animate = () => {  
36    requestAnimationFrame(animate)  
37    cube.geometry.applyMatrix4( transformation )  
38    controls.update()  
39    renderer.render( scene, camera)  
40  }  
41  animate()
```

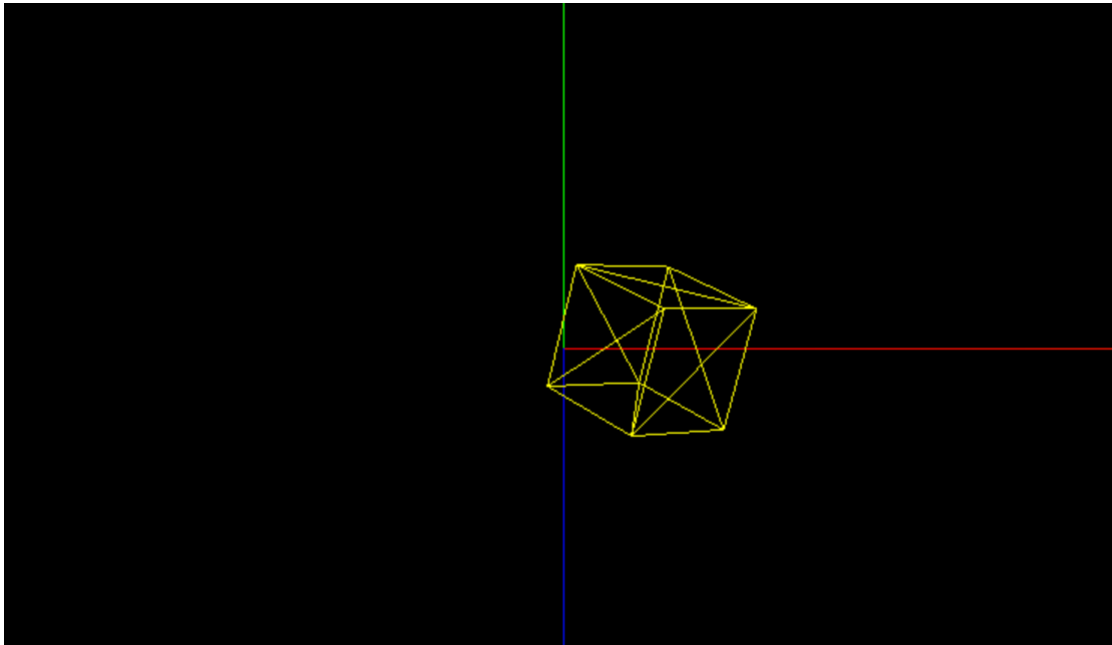


Figura 6: Gráfico\_06

## 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

```
1 var geometry = new THREE.BoxGeometry(3, 3, 3, 3, 3, 3);
2 var material = new THREE.MeshBasicMaterial({color: 0xffff00, wireframe: true});
3 var cube = new THREE.Mesh(geometry, material);
4 const translation = new THREE.Matrix4();
5 translation.set(
6     1, 0, 0, 0.02,
7     0, 1, 0, 0,
8     0, 0, 1, 0,
9     0, 0, 0, 1
10 );
11 const scala = new THREE.Matrix4();
12 scala.set(
13     1.002, 0, 0, 0,
14     0, 1.002, 0, 0,
15     0, 0, 1.002, 0,
16     0, 0, 0, 1
17 );
18 const rotaX = new THREE.Matrix4();
19 rotaX.set(
20     1, 0, 0, 0,
```

```
21     0, Math.cos(0.01), -Math.sin(0.01), 0,
22     0, Math.sin(0.01), Math.cos(0.01), 0,
23     0, 0, 0, 1
24 );
25 const rotaY = new THREE.Matrix4();
26 rotaY.set(
27     Math.cos(0.01), 0, -Math.sin(0.01), 0,
28     0, 1, 0, 0,
29     Math.sin(0.01), 0, Math.cos(0.01), 0,
30     0, 0, 0, 1
31 );
32 var ejer07 = translation.multiply(scala).multiply(rotaY).multiply(rotaX);
33 var scene = new THREE.Scene();
34 const axesHelper = new THREE.AxesHelper(5);
35 scene.add(axesHelper);
36 scene.add(cube);
37 var camera = new THREE.PerspectiveCamera(30, window.innerWidth/window.innerHeight, 0.1,
    ↪ 1000);
38 camera.position.set(20, 20, 20);
39 var renderer = new THREE.WebGLRenderer();
40 renderer.setSize( window.innerWidth, window.innerHeight );
41 document.body.appendChild( renderer.domElement );
42 var controls = new THREE.OrbitControls( camera, renderer.domElement );
43 controls.minDistance = 5;
44 controls.maxDistance = 100;
45 var animate = function () {
46     requestAnimationFrame(animate);
47     cube.geometry.applyMatrix4(ejer07);
48     renderer.render(scene, camera);
49 }
50 animate();
```

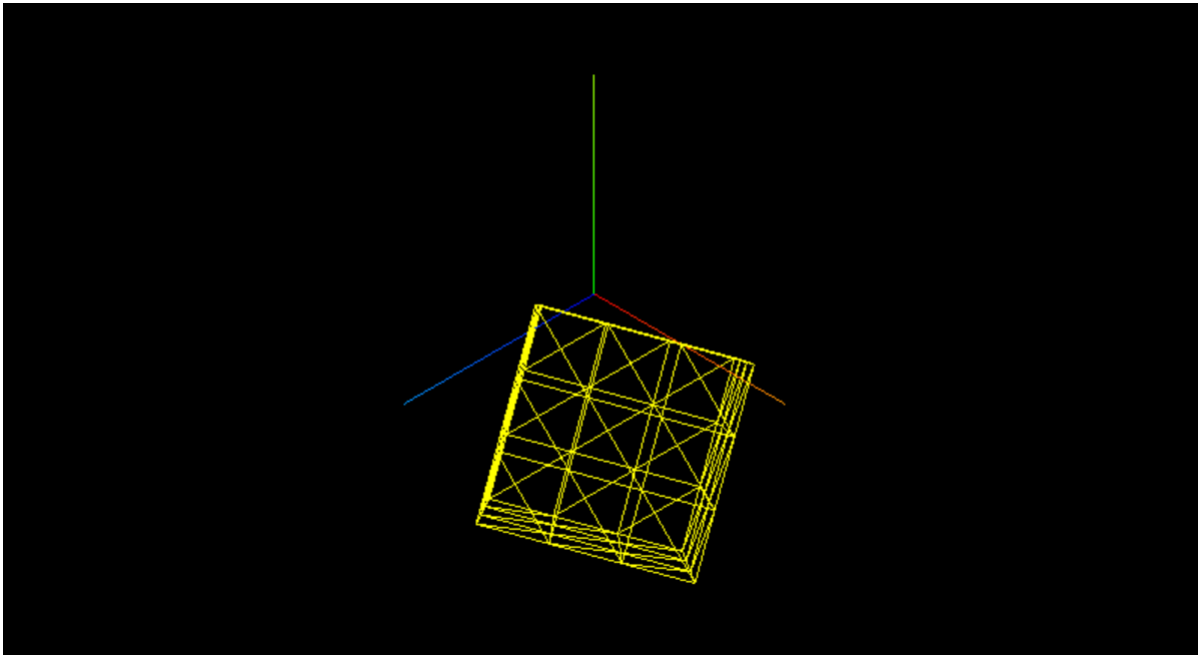


Figura 7: Gráfico\_07