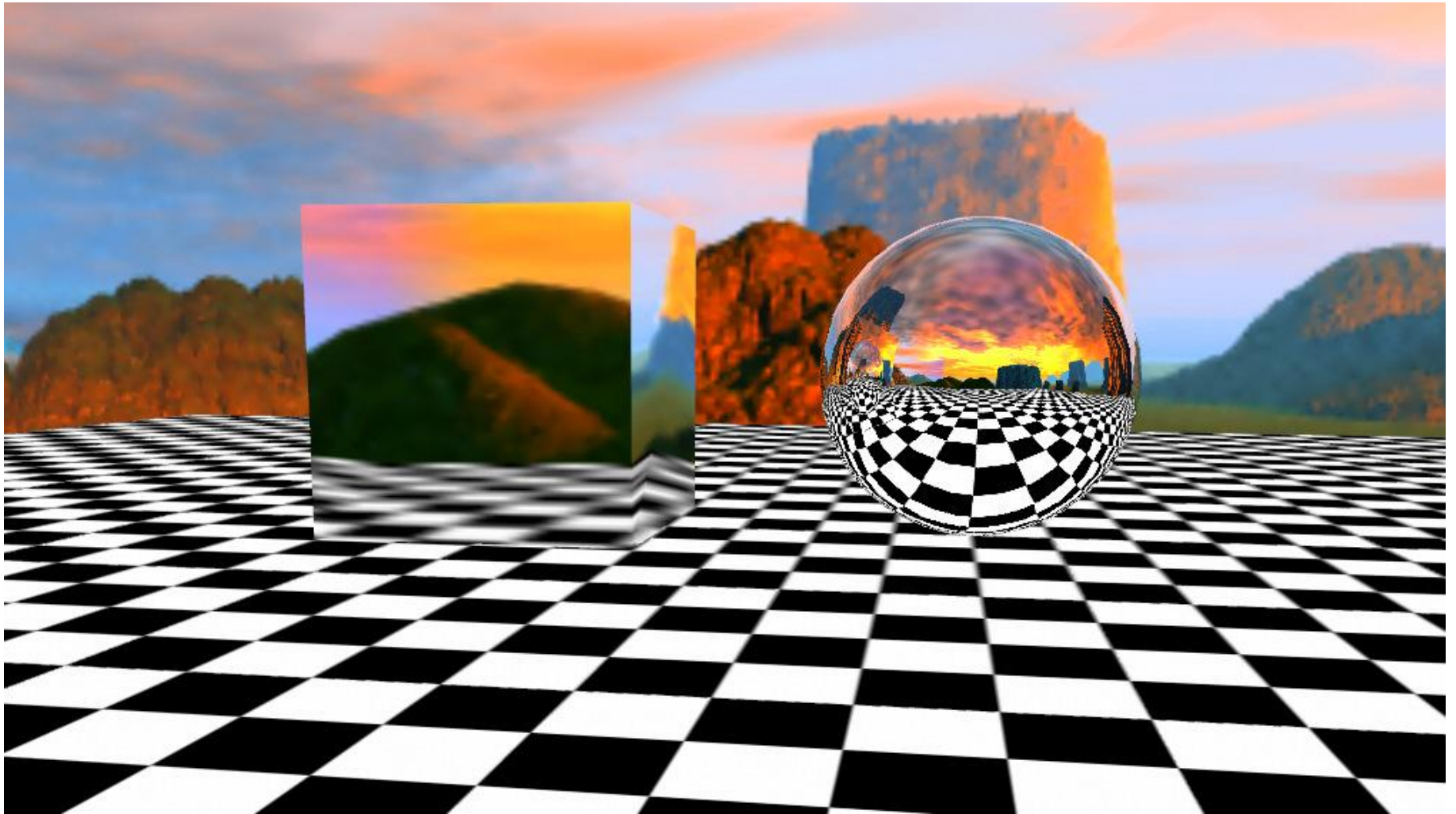


Three. Js

J. Fco. Jafet Pérez López



Se utiliza javascript para especificar una escena

```
<script type="text/javascript">
```

aquí va la Escena

```
</script>
```

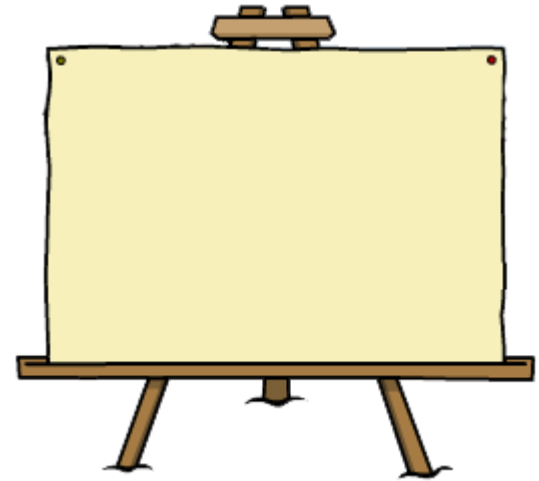


Se Requiere un elemento html como
escenario o lienzo para volcar la
escena

`<div id="escenario"> </div>`

ó

`<canvas> </canvas>`



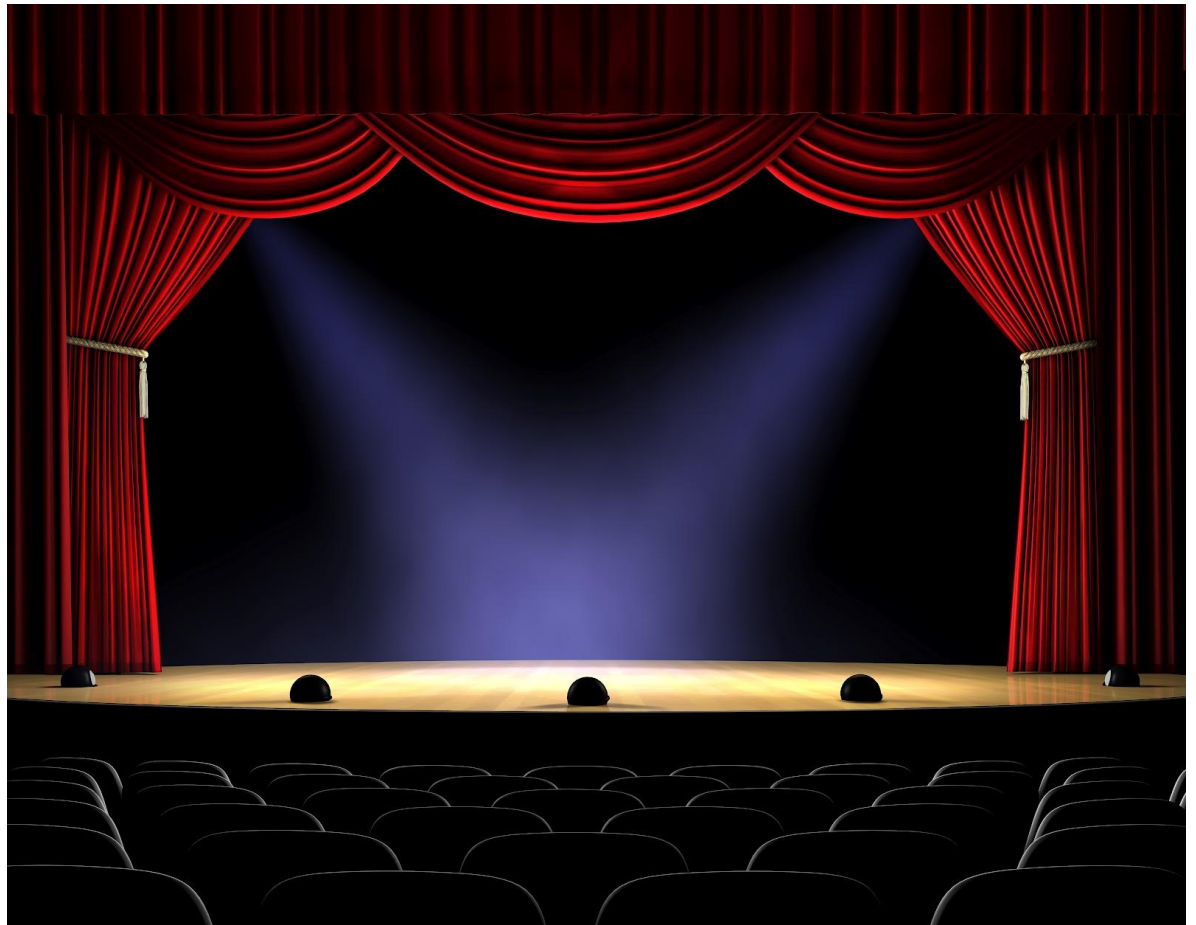
Se necesitan versiones mínimas de jquery y three.js

```
<script type="text/javascript" src='jquery-1.8.2.min.js'></script>
```

```
<script type="text/javascript" src='Three.js'></script>
```

Variables para las dimensiones de la escena

```
var ancho = 800,  
    alto  = 600;
```

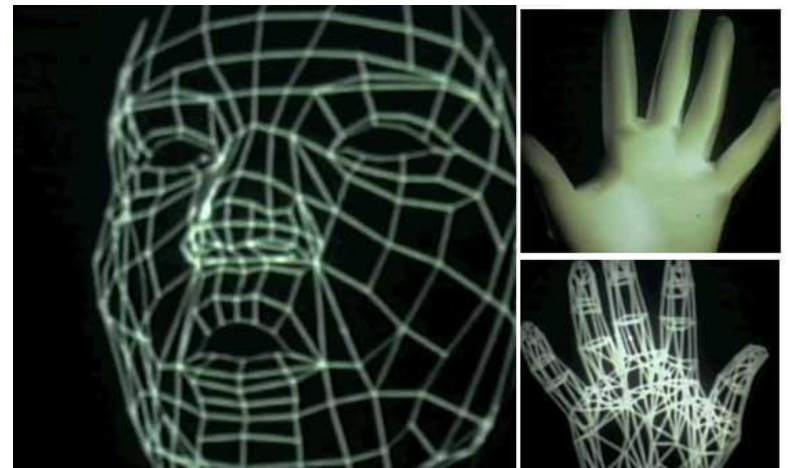


Objeto para renderizado

```
var miRender = new THREE.WebGLRenderer();
```

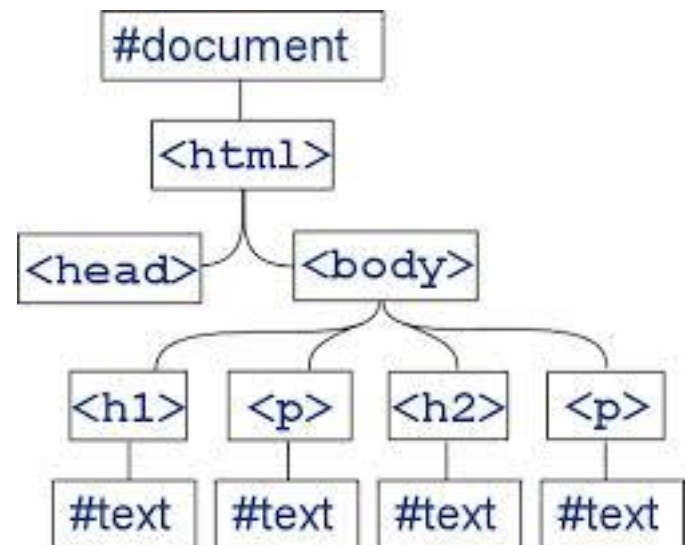
```
miRender.shadowMapEnabled = true;
```

```
miRender.setSize(ancho, alto);
```



Objeto para acceder al elemento html

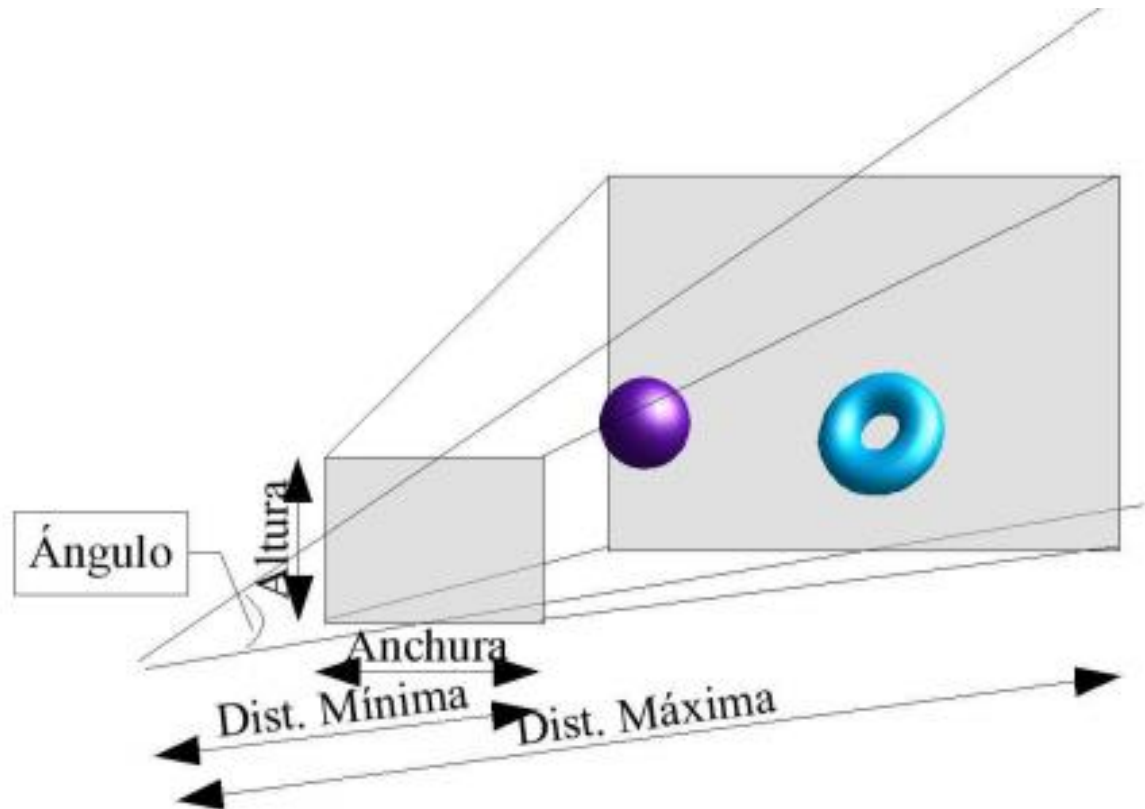
```
var escenario = $('#escenario');  
escenario.append(miRender.domElement);
```



La cámara

La proyección perspectiva:

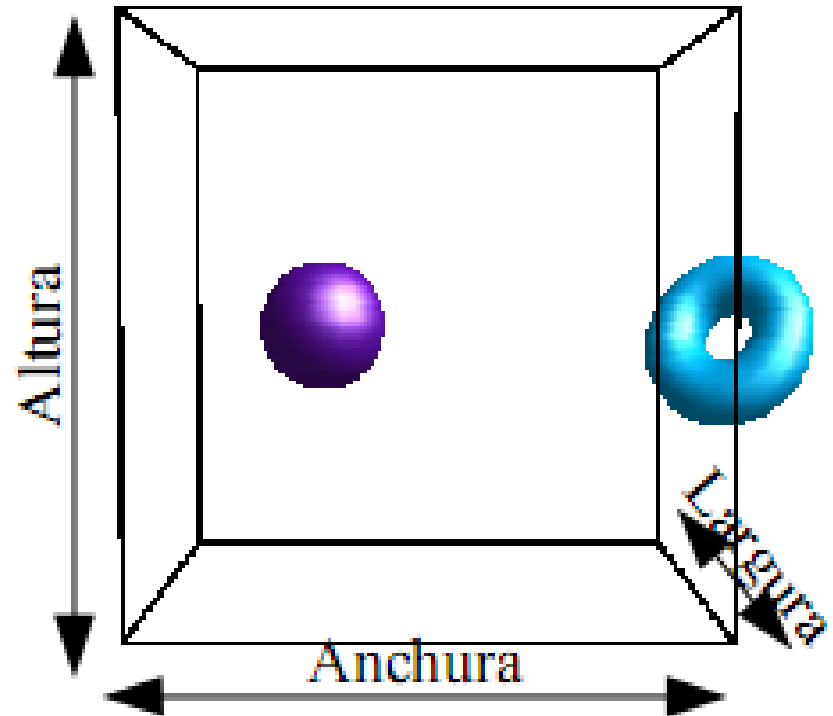
La proyección perspectiva delimita un volumen de visualización dado por un ángulo de cámara, y una relación alto/ancho. La distancia al observador determinará el tamaño con el que un objeto se visualiza.



La cámara

La proyección ortográfica:

La proyección ortográfica nos permite visualizar todo aquello que se encuentre dentro de un cubo, delimitado por ciertos parámetros. A la hora de visualizar, la distancia al observador sólo se tiene en cuenta para determinar si el objeto está dentro o fuera del cubo...



Configurando la cámara

```
var angulo = 30,  
    ratio = ancho / alto,  
    cerca = 0.1,  
    lejos = 10000;
```



```
var miCamara = new THREE.PerspectiveCamera(angulo, ratio, cerca, lejos);
```

```
miCamara.position.x = 300;  
miCamara.position.y = 300;  
miCamara.position.z = -300;
```

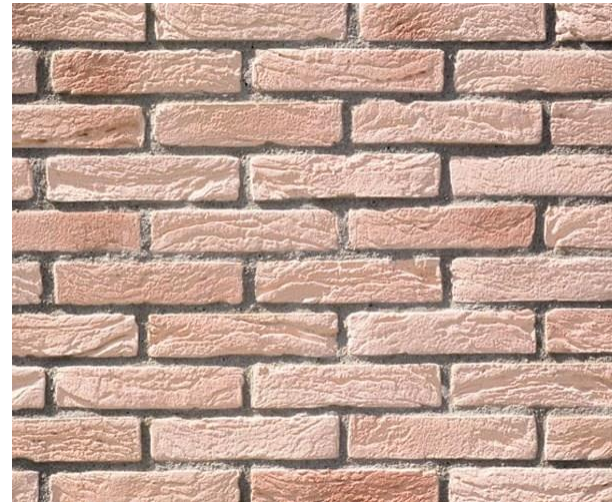
Objeto para la escena

```
var miEscena = new THREE.Scene();
```



Materials

```
var material1 = new THREE.MeshLambertMaterial({color: 0x00ff00});  
var material2 = new THREE.MeshLambertMaterial({color: 0x0000ff});
```



Agregando un cubo

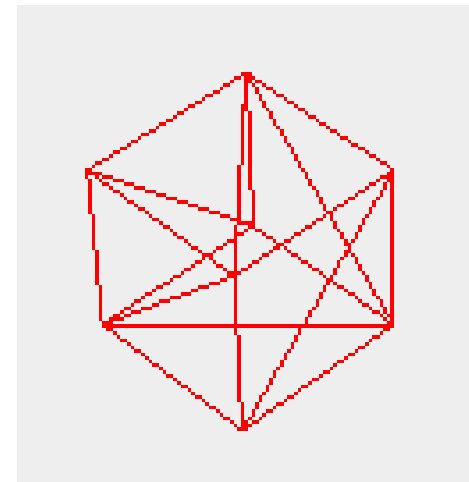
```
var cubo = new THREE.Mesh( new THREE.CubeGeometry(60, 60, 60), material1);
```

```
cubo.position.x = 0;
```

```
cubo.position.y = 40;
```

```
cubo.position.z = 0;
```

```
cubo.receiveShadow = true;
```



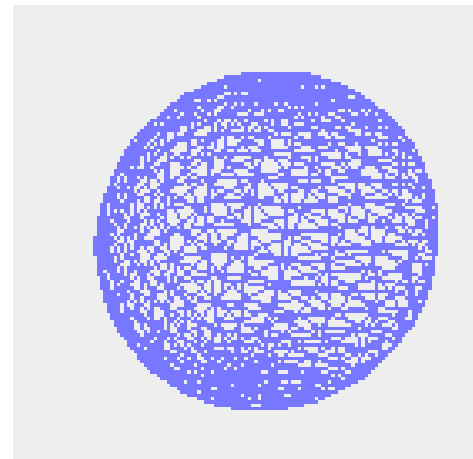
Agregando una esfera

```
var radio = 50,  
    segmentos = 16,  
    anillos = 16;
```

```
var esfera = new THREE.Mesh( new THREE.SphereGeometry(radio, segmentos, anillos), material2);
```

```
esfera.position.x = 100;  
esfera.position.y = 40;  
esfera.position.z = 0;
```

```
esfera.receiveShadow = true;
```



Agregando fuentes de luz

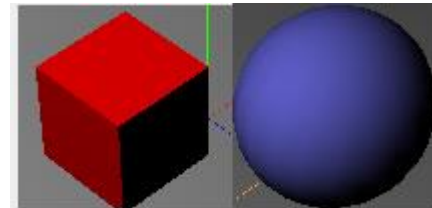
```
var luz = new THREE.SpotLight(0xFFFFFFFF);
```

```
luz.position.x = 0;
```

```
luz.position.y = 200;
```

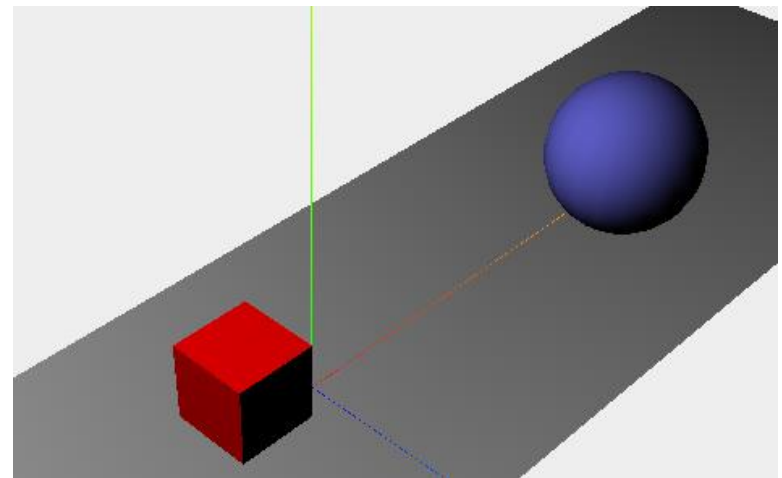
```
luz.position.z = 200;
```

```
luz.castShadow = true;
```



Armando la escena

```
miEscena.add(luz);  
miEscena.add(cubo);  
miEscena.add(esfera);  
miEscena.add(miCamara);
```



Apuntando la cámara

```
miCamara.lookAt(cubo.position);
```



Renderizando la escena

```
miRender.render(miEscena, miCamara);
```

