**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

****

**INTEGRANTES:**

**Karen Lilibeth Borbor Moreira**

**Ivan Alejandro Mera Maldonado**

**José Antonio Viteri Cuenca**

**PROYECTO:**

**Creación de figuras geométricas e implementación de luces y sombreado**

Descripción del proyecto

**TreeeJs**

[Three.js](http://threejs.org/) es una librería bastante liviana y muy eficiente para generar y animar gráficos en 3D dentro del navegador, aprovechando las grandes novedades que nos ofrece HTML5 para la generación de contenidos multimedia. Aprovecha tanto las capacidades de HTML5 que es capaz de generar escenas 3D con WebGL, Canvas (2D) y SVG.

**Proceso de creación del proyecto**

Antes de poder utilizar Three.js, se necesita un lugar para mostrarlo. Para esto se realiza un archivo html con un container, el cual será en donde se mostrará la pantalla de three.js.

<html>

<head>

<title>Proyecto</title>

<script type="text/javascript" src="http://code.jquery.com/jquery.min.js"></script>

<script src="lib/three.min.js"></script>

<script src="lib/DragControls.js"></script>

<script type="text/javascript" src="js/Detector.js"></script>

</head>

<body>

<div id="canvas"></div>

<script src="js/principal.js"></script>

</body>

</html>

### **Creando la escena**

Para interactuar con Three.js lo haremos a través del objeto THREE, que nos brinda todos los elementos necesarios.

Lo primero de todo es generar el renderizador, que en este caso usaremos el de WebGL, que es el más completo. Posteriormente le daremos las dimensiones de la pantalla y lo añadiremos al container.

// Instanciamos el escenario. Este va a contener todas las figuras y elementos.

scene = new THREE.Scene();

// Instanciamos el renderizador. Nos permitirá construir el escenario junto a las figuras que le hayamos agregado.

renderer = new THREE.WebGLRenderer();

// Para configurar el renderizador debemos indicar por lo menos el alto y ancho al que queremos que renderice la app, y luego añadirlo al DOM.

renderer.setPixelRatio(window.devicePixelRatio);

renderer.setSize(window.innerWidth, window.innerHeight);

renderer.setClearColor(0x000000, 1);

$('#canvas').append(renderer.domElement);

renderer.shadowMap.enabled = true;

renderer.shadowMap.type = THREE.PCFSoftShadowMap;

renderer.gammaInput = true;

renderer.gammaOutput = true;

// También podemos añadir los ejes de coordenadas al escenario

var sceneAxis = new THREE.AxisHelper(100);

scene.add(sceneAxis);

### **Iluminación de la escena**

Seguidamente se creo un objeto para la iluminación y su posterior renderizado en la escena con para eso utilizamos una spotlight con la siguiente función:

// Existen algunos tipos de luz.

// La luz de tipo 'SpotLight' se emite desde una determinada dirección y permite generar sombras.

spotLight = new THREE.SpotLight( 0xffffff, 1 );

spotLight.position.set(-25, 30, 5);

// spotLight.angle = Math.PI / 4;

spotLight.penumbra = 0.05;

spotLight.decay = 2;

spotLight.distance = 200;

spotLight.castShadow = true;

spotLight.shadow.mapSize.width = 1024;

spotLight.shadow.mapSize.height = 1024;

spotLight.shadow.camera.near = 10;

spotLight.shadow.camera.far = 200;

scene.add(spotLight);

// También podemos instanciar helpers, que nos permitirán dibujar el área de acción de la luz o la sombra en el escenario

lightHelper = new THREE.SpotLightHelper( spotLight );

scene.add( lightHelper );

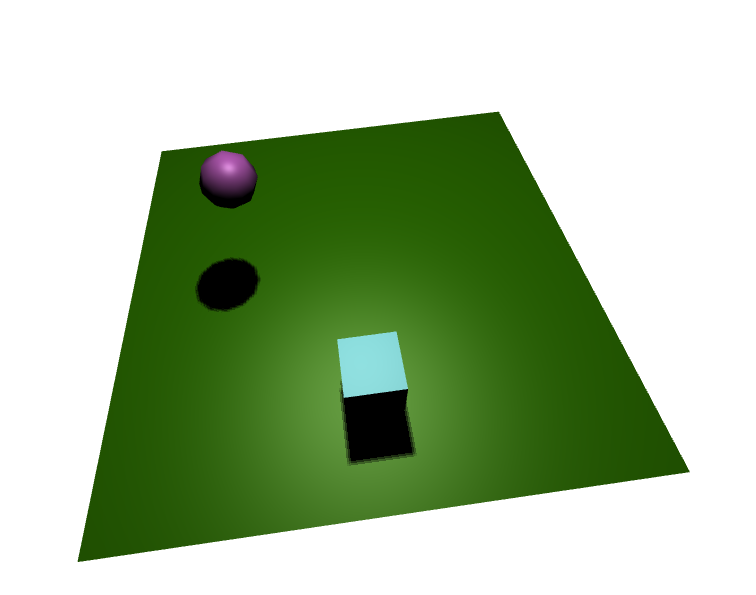
// La luz de tipo 'AmbientLight' ilumina de igual manera a todos los objetos de la escena, pero no genera sombras

var ambient = new THREE.AmbientLight( 0xffffff, 0.025 );

scene.add( ambient );

### **Creación de objetos geométricos**

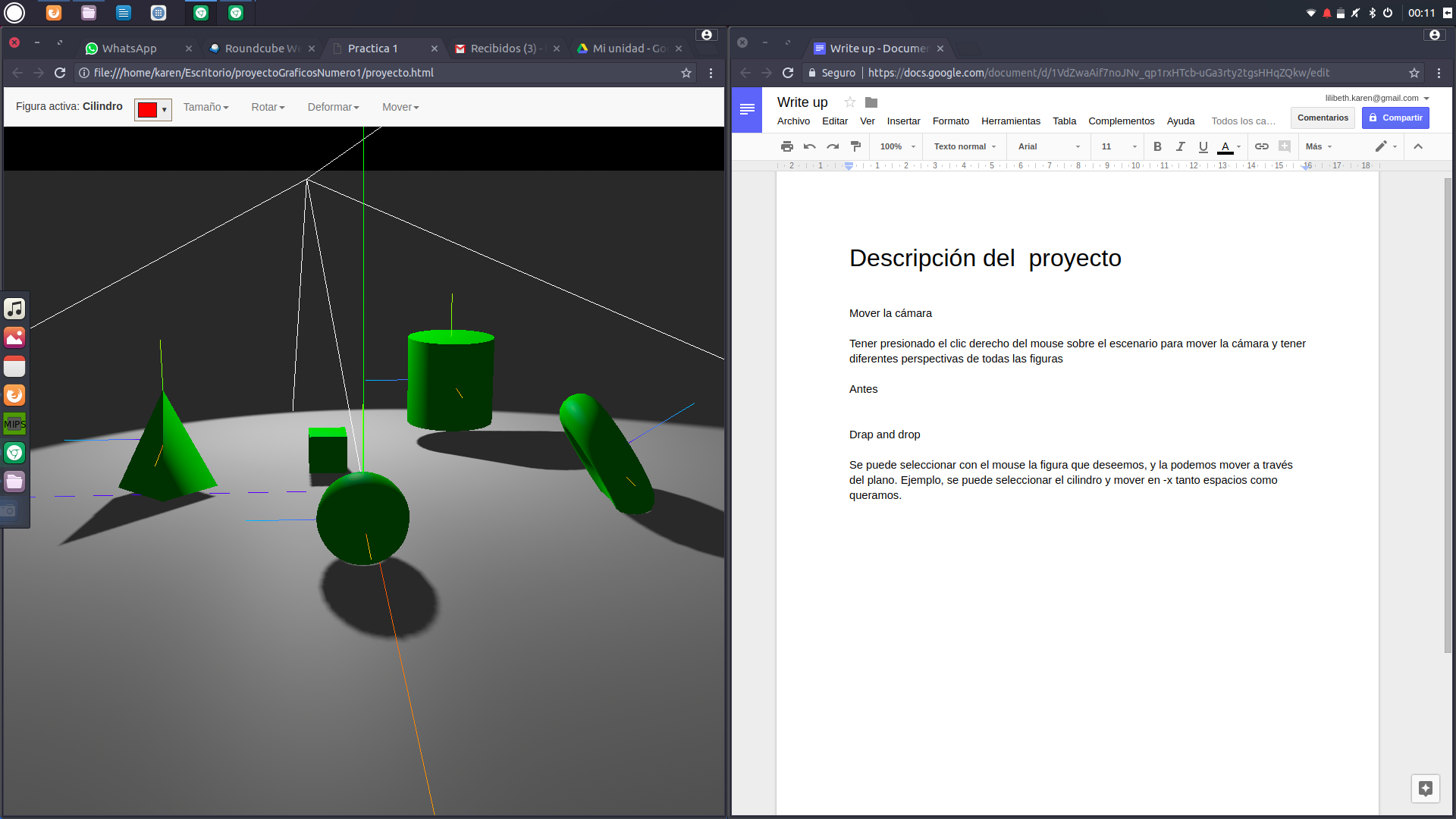
Se crean las figuras solicitadas como lo son el cubo, la esfera, el toroide, el cilindro, pirámide para luego ser colocadas a la escena



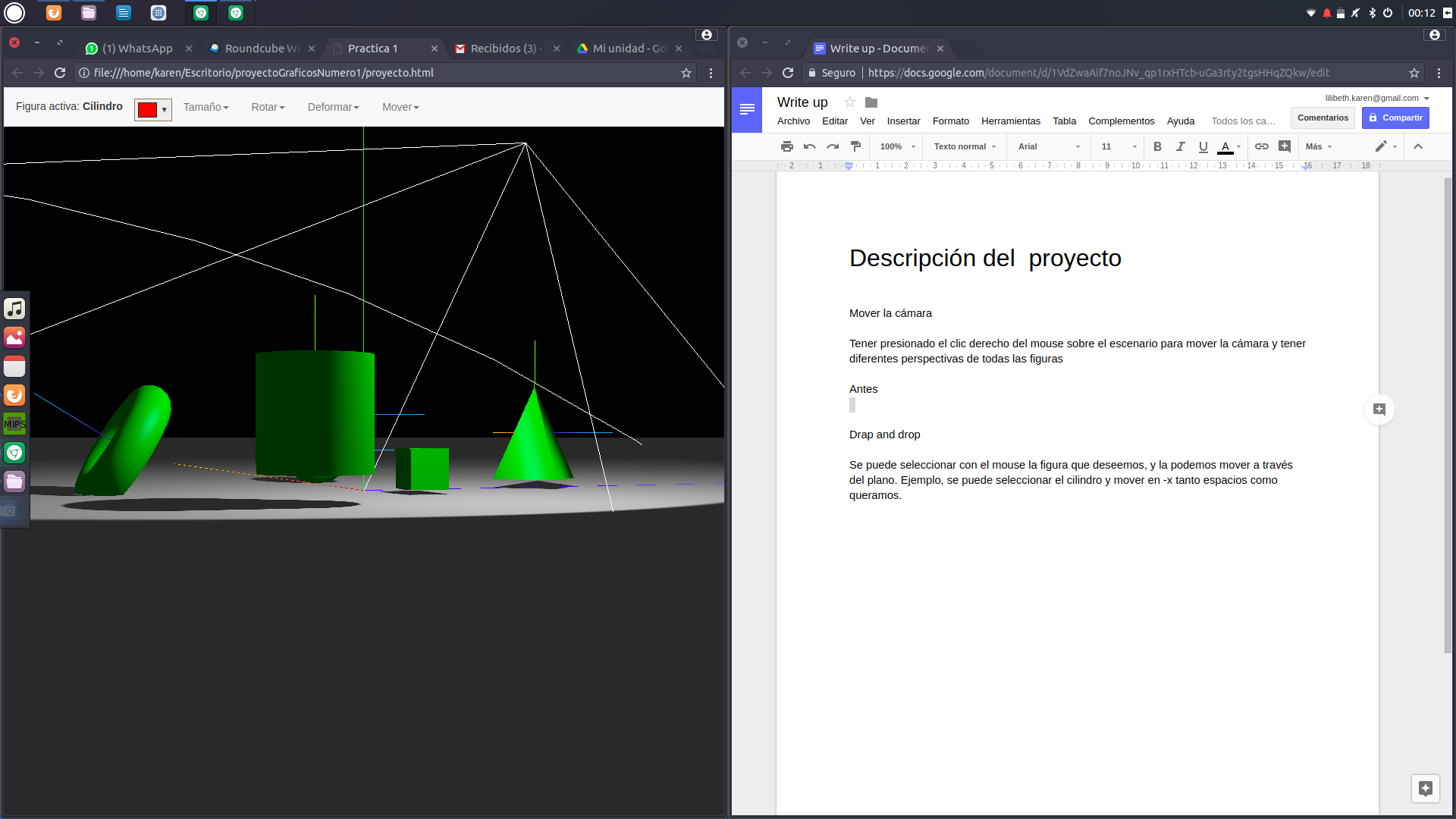
**Mover la cámara**

Tener presionado el clic derecho del mouse sobre el escenario para mover la cámara y tener diferentes perspectivas de todas las figuras

Antes



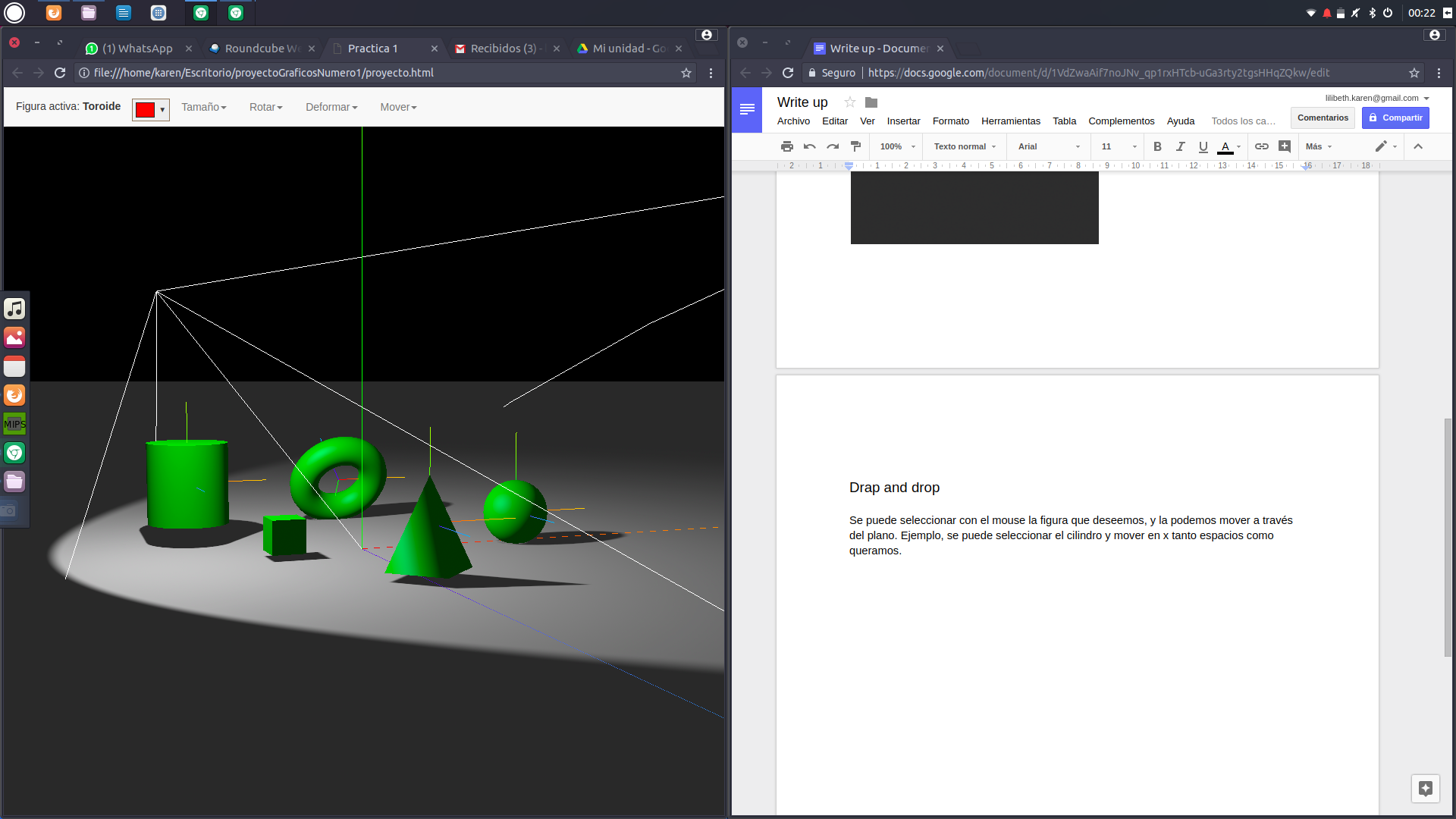
Después



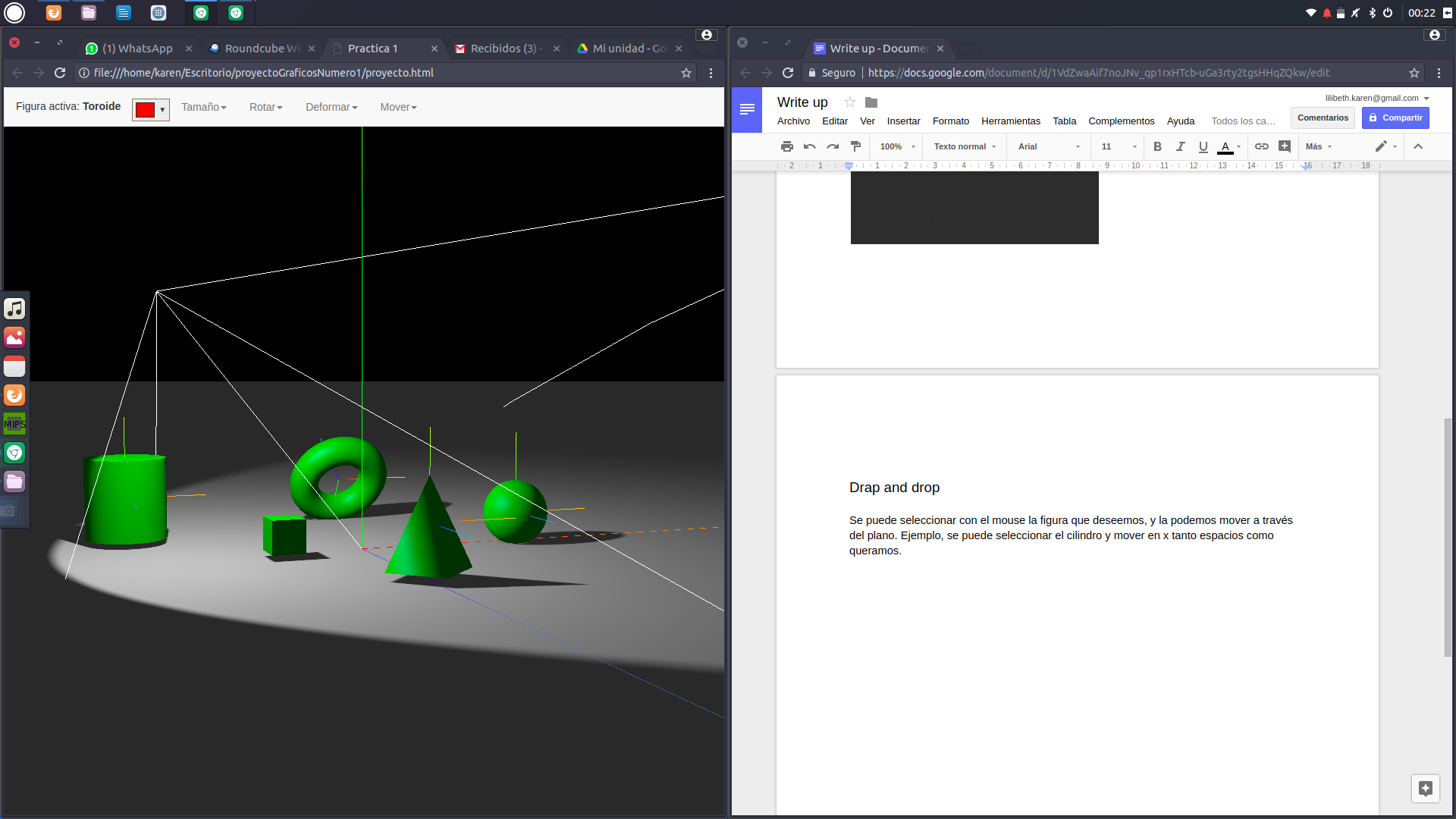
**Drap and drop**

Se puede seleccionar con el mouse la figura que deseemos, y la podemos mover a través del plano. Ejemplo, se puede seleccionar el cilindro y mover en x tanto espacios como queramos.

Antes



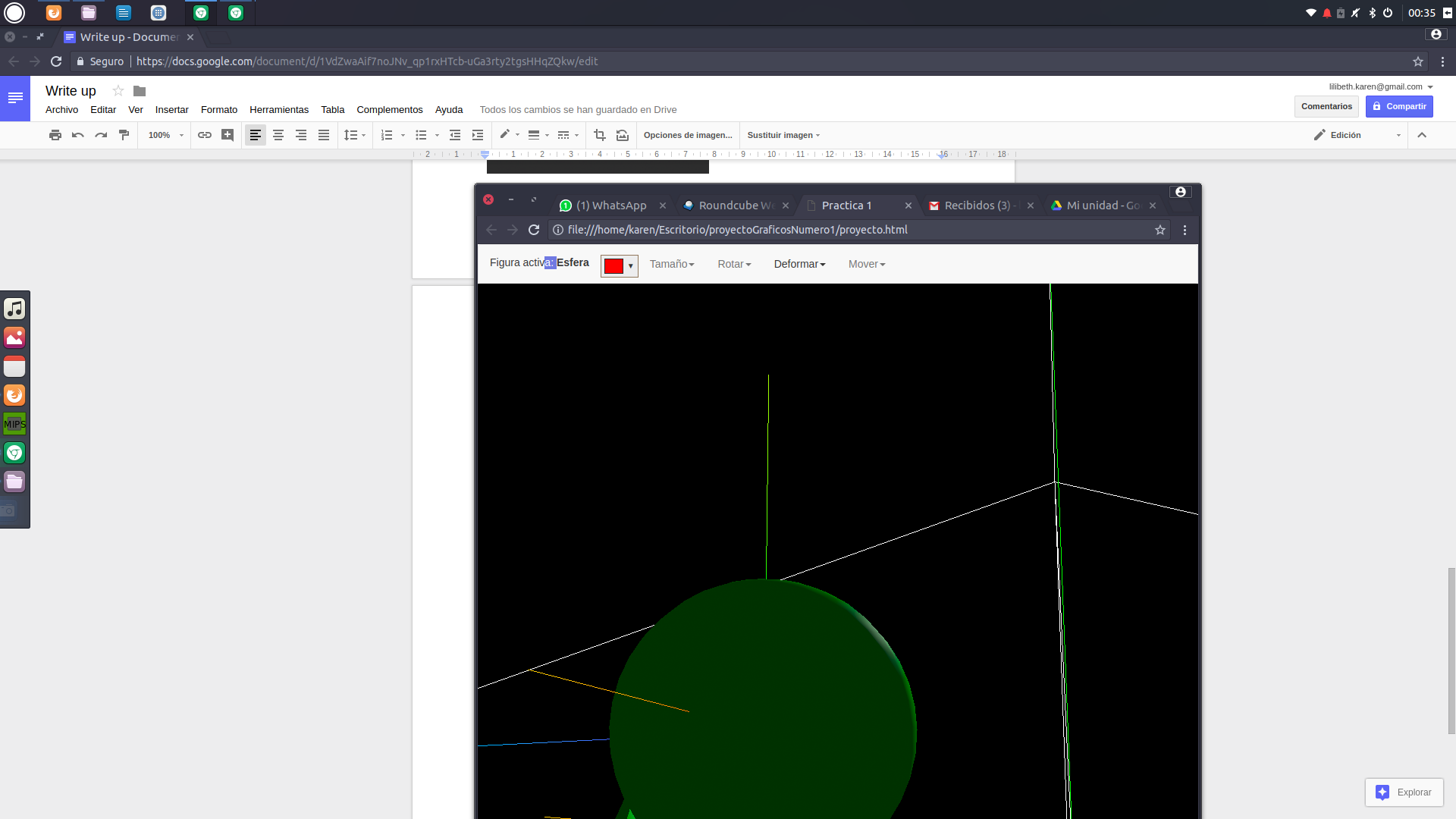
Después



**Mostrar la figura activa**

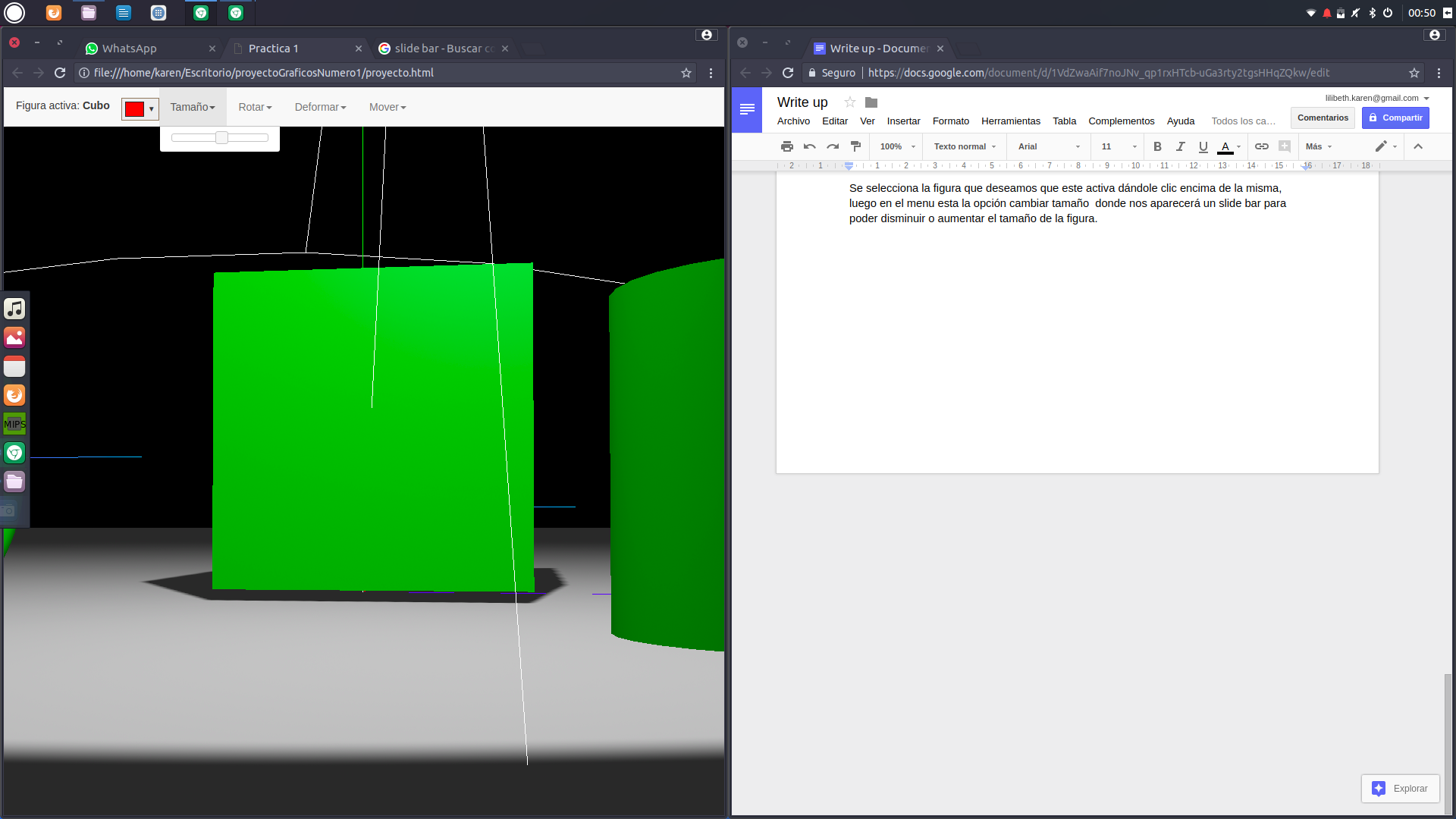
Se muestra en el menú la figura que está activa, esto significa que las acciones que se deseen realizar se van a realizar sobre ella. Las acciones que se le puede realizar son cambiar el tamaño de la figura, rotarla, deformarla y moverla.

Para cambiar la figura activa se lo puede hacer dando clic encima de la figura al cual queremos que esté activa.



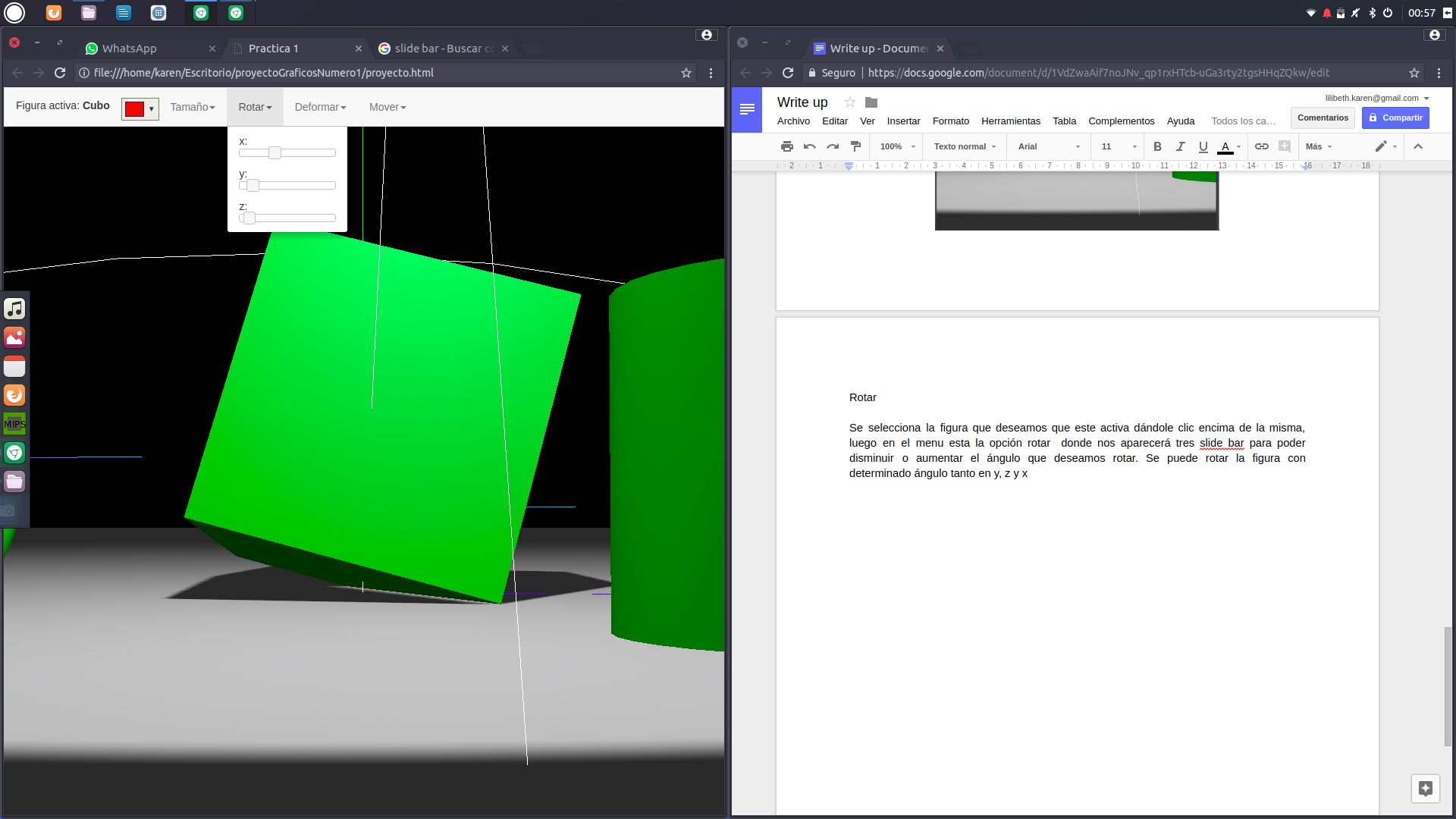
**Cambiar el tamaño**

Se selecciona la figura que deseamos que este activa dándole clic encima de la misma, luego en el menu esta la opción cambiar tamaño donde nos aparecerá un slide bar para poder disminuir o aumentar el tamaño de la figura



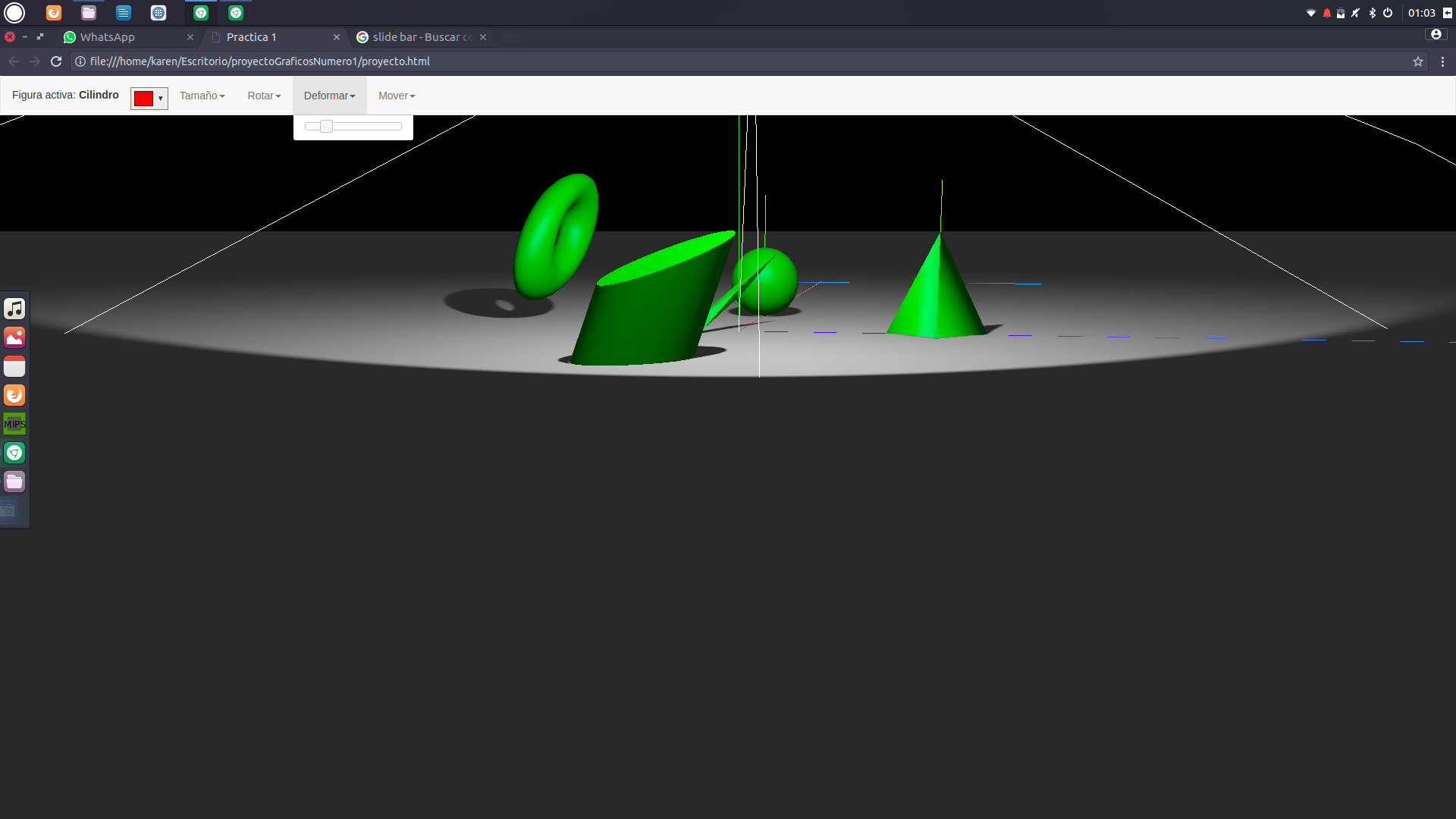
**Rotar**

Se selecciona la figura que deseamos que este activa dándole clic encima de la misma, luego en el menu esta la opción rotar donde nos aparecerá tres slide bar para poder disminuir o aumentar el ángulo que deseamos rotar. Se puede rotar la figura con determinado ángulo tanto en y, z y x



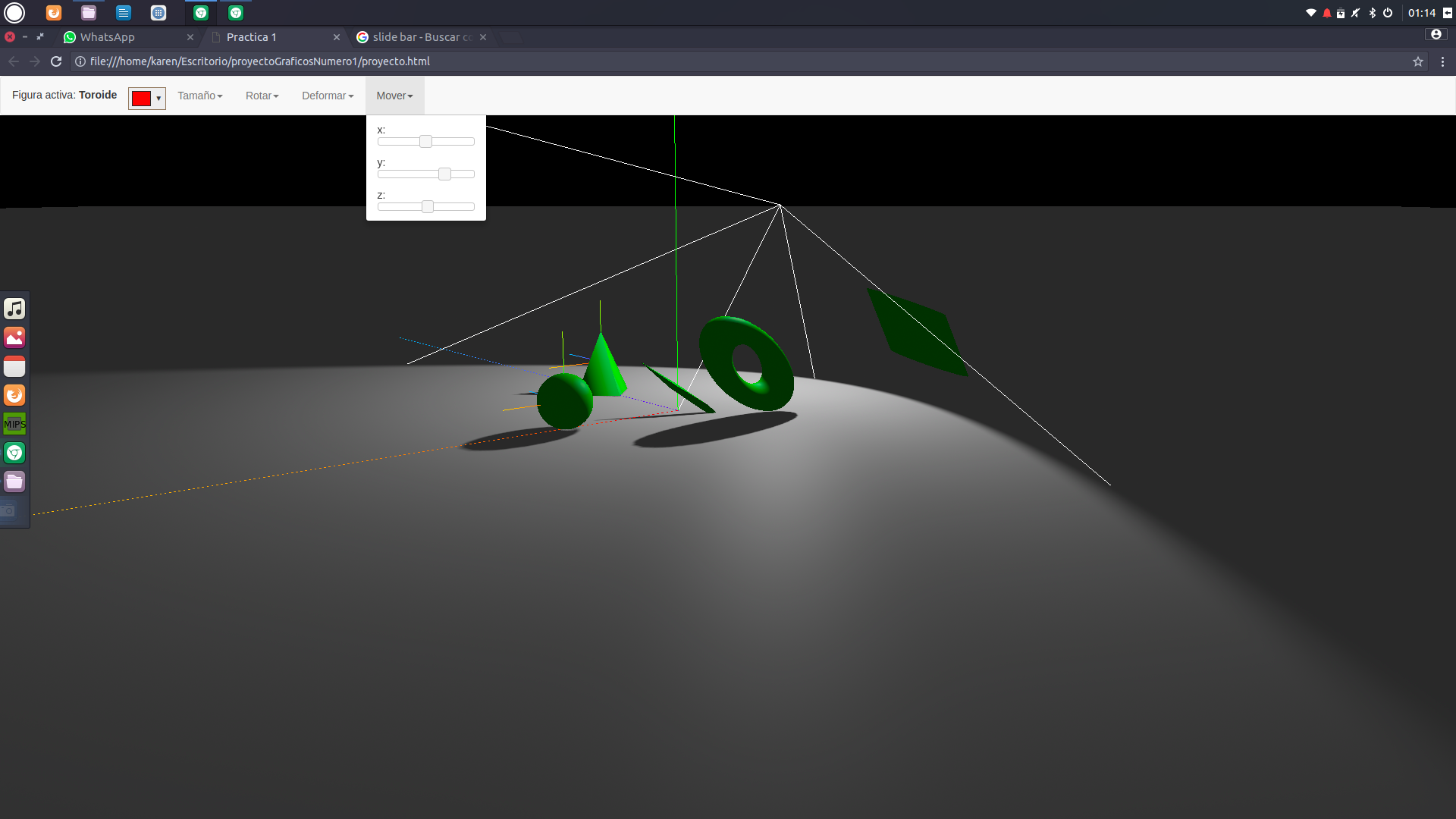
**Deformar**

Se selecciona la figura que deseamos que este activa dándole clic encima de la misma, luego en el menu esta la opción deformar donde nos aparecerá un slide bar para poder disminuir o aumentar que tanto queremos deformar la figura. Donde cuando tiene un 0 por ciento es la figura en su forma natural, cuando esta el 100 por ciento se deforma en su totalidad. Por ejemplo, en la siguiente imagen se deformo el cilindro



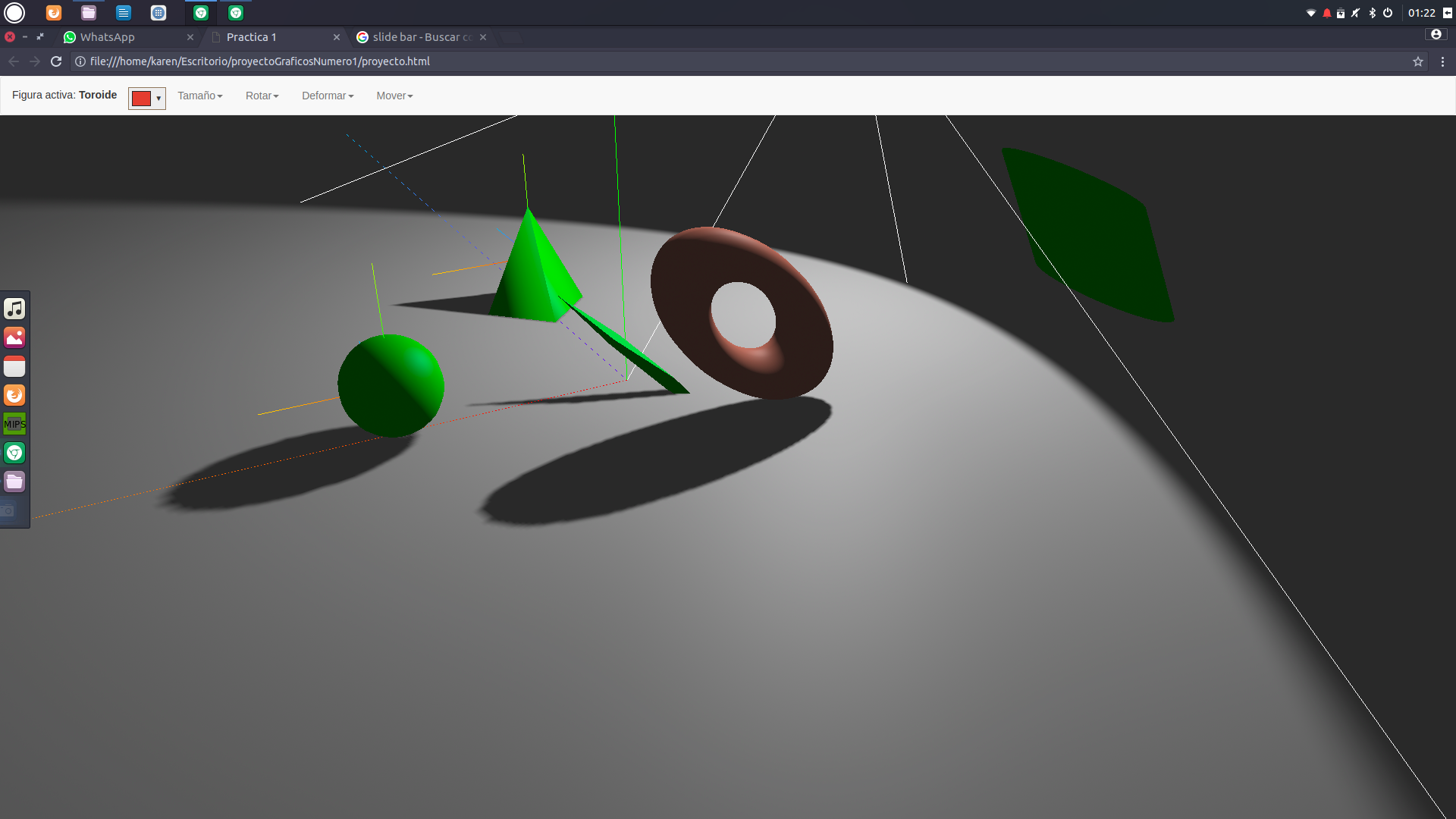
**Mover**

Se selecciona la figura que deseamos que este activa dándole clic encima de la misma, luego en el menu esta la opción mover donde nos aparecerá tres slide bar para poder disminuir o aumentar que tanto se desea mover la figura. Por ejemplo, en la siguiente imagen se movió el toroide tanto en x, z y y a través del plano



**Cambiar el color**

Se selecciona la figura que deseamos que este activa dándole clic encima de la misma, luego en el menu esta la opción cambiar color donde nos aparecerá un selector de color donde podemos elegir el color que deseamos.



**Implementación de picking**

Se implementó una función llamada onMouseClick, la cual manera las coordenadas del mouse en el lugar que se dio click para poder realizar las diferentes opciones que se muestra en el menú .