



## **Manual de prácticas de laboratorio**

Título: **Graficación 2d y 3d**

Para la Unidad de Aprendizaje: **Graficación**

Créditos de la Unidad de Aprendizaje: **5**

Espacio académico en que se imparte la UA: **CU Valle de Chalco**

Versión del plan de estudios 2005

### **Datos de Identificación**

Programa Educativo: **Ingeniería en Computación**

Programa de Estudios por Competencias :**Graficación**

Unidad de Competencia: 1- 6.

Elaborado por:

**Autores:**

**M.T.E. Marisol Hernández Hernández**

**M. en T.I. Rodolfo Melgarejo Salgado**

**M. en I. Marco Alberto Mendoza Pérez**

Junio del 2019



## Índice

Datos de Identificación	1
Presentación	3
Estructura de la Unidad de Aprendizaje	4
Propósito de la Unidad de Aprendizaje	4
Prácticas:	
Dibujando imágenes 2D	5
Traslación, Escalado, Rotación y Reflexión.	8
Crear modelos poligonales y poliédricos.	11
Modelar objetos 2d	13
Modelar objetos 3d	16
Luces	21
Animación 3D	24
Referencias	29





## Presentación

La generación y representación de imágenes infográficas implica la reducción de un mundo tridimensional a una pantalla bidimensional. Esta disminución en la dimensionalidad conforma un mapeado muchos-a-uno. Los cambios que ocurren en 3D deben ser correctamente modelados matemáticamente para poder calcular su proyección sobre la pantalla. El futuro ingeniero encontrará esta unidad de aprendizaje interesante debido a las múltiples aplicaciones que se abren en el mundo laboral: infografía, animación, generación de efectos especiales visuales para cine y televisión, realidad virtual, modelado de piezas y estructuras, diseño asistido por ordenador (CAD), software de ocio, motores 3D y un largo etcétera.

La estructura planteada consta de seis unidades de competencia. La primera se centra en los conceptos de geometría que serán base para el resto de unidades de competencia. La segunda estudia las transformaciones básicas de 2D y de encuadre y perspectiva. La tercera concentra todo lo relativo al modelado tanto poligonal como poliédrico. La cuarta continúa con el modelado de objetos en 2D y 3D. La quinta unidad de competencia reúne los conocimientos acerca de la iluminación de la escena y la visualización de objetos. La sexta entra en temas avanzados como el fotorrealismo o la animación.

Para esta unidad de aprendizaje y considerando que a la semana son 3 horas, de las cuales 2 son de teoría y 1 de práctica, por lo que se considera 1 práctica por cada unidad y tomando en cuenta que la asignatura contempla tanto el diseño de imágenes bidimensionales, tridimensionales y animación, se abarcarán esos aspectos en las prácticas realizadas.

Dentro de las prácticas propuestas de toma en cuenta los siguientes aspectos:

La **primera práctica** se centra en geometría y con ayuda del lenguaje HTML5 los alumnos aprenden los conceptos como 2d, vectores, coordenadas, plano cartesiano, etc., que son elementales para poder comprender las prácticas siguientes.

La **segunda práctica** aplica las transformaciones básicas de 2D y de encuadre y perspectiva aplicando a los trazos 2d los conceptos de traslación, rotación y reflexión.

Para la **tercera práctica**, ya se están aplicando los conceptos de imágenes poligonales y poliédricas, haciendo uso del programa de KetchUp Free que es el software de modelado 3D gratuito más simple de la web, sin condiciones (<https://www.sketchup.com/es/plans-and-pricing/sketchup-free>).

Para la **cuarta unidad se abarcan 2 prácticas**, una para modelar imágenes 2d en HTML, aplicando los conceptos vistos en las prácticas anteriores y otra modelando ya objetos 3d en SketchUp.

La **quinta unidad** de competencia se aplican la iluminación de escena y visualización de objetos, mismo que se hacen con ayuda de los colores, que simulan la luz en las escenas.

La **sexta unidad** toca el tema de animación, la cual mediante movimientos de la cámara y de los objetos crea escenas animadas.

Es importante especificar que estas prácticas ya se implementaron con los alumnos de ingeniería en computación en la unidad de aprendizaje “**Graficación**”.





### Estructura de la Unidad de Aprendizaje

1. Encuadrar la graficación en el mercado, comprender conceptos de geometría espacial y ser perito en el cálculo vectorial.
2. Aplicar transformaciones básicas elementales en 2D y transformaciones de encuadre y perspectiva
3. Modelar objetos poligonales y poliédricos
4. Modelar objetos en 2D y 3D.
5. Visualizar objetos de una escena en la pantalla controlando las condiciones lumínicas
6. Generar imágenes por computador que emulen a la realidad y animar los objetos presentes en la escena añadiéndoles movimiento.

#### Programa de Estudios por Competencias

### Graficación

#### I. IDENTIFICACIÓN DEL CURSO

<b>ORGANISMO ACADÉMICO:</b> Facultad de Ingeniería de la Universidad Autónoma del Estado de México (UAEMex)								
<b>Programa Educativo:</b> Ingeniería en Computación				<b>Área de docencia:</b> Interacción Hombre-Máquina				
<b>Aprobación por los H.H. Consejos Académico y de Gobierno</b>			<b>Fecha:</b>		<b>Programa elaborado por:</b> Dr. Felipe Orihuela Espina Dra. Adriana Vilchis González		<b>Fecha de elaboración :</b> 20-Noviembre-2005	
Clave	Horas de teoría	Horas de práctica	Total de horas	Créditos	Tipo de Unidad de Aprendizaje	Carácter de la Unidad de Aprendizaje	Núcleo de formación	Modalidad
L41064	2	1	3	5	Curso y laboratorio	Obligatoria	Sustantivo	Presencial
<b>Prerrequisitos ( Conocimientos Previos):</b> Ninguno					<b>Unidad de Aprendizaje Antecedente</b> Ninguna		<b>Unidad de Aprendizaje Consecuente</b> Ninguna	
<b>Programas educativos en los que se imparte:</b> UASP: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacan, Valle de Zumpango								

#### Propósito

Encuadrado en el Plan Flexible 2004 por Competencias de la UAEMex, presentar al alumno los fundamentos y algoritmos básicos de uso cotidiano en el campo de infografía y grafización y presentarle el campo de generación de imágenes como uno de los campos con aplicaciones potenciales más variadas en el mercado de trabajo.







## Unidad de competencia I.

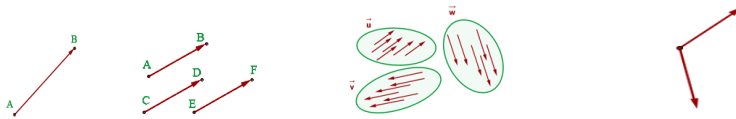
Encuadrar la graficación en el mercado, comprender conceptos de geometría espacial y ser perito en el cálculo vectorial.

### Práctica 1: Dibujando imágenes 2D

#### Introducción

**Sistema de coordenadas.** Es un conjunto de valores y puntos que permiten definir unívocamente la posición de cualquier punto de un espacio, se suele referir a ellas como coordenadas cartesianas. En este plano se representa un punto en un plano, utilizando dos rectas perpendiculares entre sí, de forma que la posición del punto se determina midiendo sobre los ejes las distancias al punto (Mejía, Jiménez y Fernández, 2005).

Un espacio vectorial (o espacio lineal) es el objeto básico de estudio en la rama de la matemática llamada álgebra lineal, sus elementos son vectores, sus vectores pueden ser fijos que son segmentos orientados que van del punto A (origen) al punto B (extremo), también se tienen vectores equipolentes, cuando tienen el mismo módulo, dirección y sentido, o vectores libres y vectores concurrentes (Navarro, 2009). Tipos de líneas: rectas, quebradas, onduladas, rizadas, circunferencias, espirales, en forma de ocho.



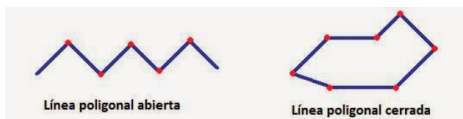
**Línea poligonal cerrada:** la que se forma cuando unimos segmentos de recta de un plano, en la que los segmentos extremos no coinciden en un mismo punto.

**Línea poligonal abierta:** la que se forma cuando unimos segmentos de recta de un plano y en la que los segmentos extremos coinciden en un mismo punto.

**Línea curva cerrada:** línea recta que toma algún tipo de desviación en su rectitud de manera progresiva

**Línea recta:** sucesión continua e indefinida de puntos en una sola dimensión.

•



•



•

#### Descripción de la práctica

Crear figuras utilizando vectores en 2d. Se crearán scripts en JavaScript y HTML5.

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Para dibujar gráficos 2D se utiliza canvas, que es un elemento HTML que a través de JavaScript, funciona como un contenedor para gráficos, así el lenguaje JavaScript proporciona varios métodos para dibujar trazados, cuadros, círculos, texto y agregar imágenes.

Duración: 1 hora

**Objetivos de aprendizaje:** El alumno aprenderá a dibujar en 2d, aplicando herramientas coordenadas, vectores y espacio vectorial.





### Requisitos Previos

Saber de los conocimientos básicos como: vectores, coordenadas, plano cartesiano, además saber lenguaje HTML5

### Materiales

Computadora

Navegador

Editor de texto (puede ser sublime o Notepad++)

### Procedimiento

1. Ejecutar Sublime o Notepad++
2. Crear un archivo con extensión HTML
3. Copiar el siguiente código

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>Título de la página</title>
5   <meta charset="UTF-8">
6
7 <script type="text/javascript">
8   function retornarLienzo(x) {
9     var canvas = document.getElementById(x);
10    if (canvas.getContext) {
11      var lienzo = canvas.getContext("2d");
12      return lienzo;
13    } else
14      return false;
15  }
16
17  function dibujar() {
18    var lienzo=retornarLienzo("lienzo1");
19    if (lienzo) {
20      lienzo.beginPath();
21      lienzo.strokeStyle="rgb(255,87,51)";
22      lienzo.lineWidth=7;
23      lienzo.moveTo(10,5);
24      lienzo.lineTo(10,295);
25      lienzo.stroke();
26
27      lienzo.beginPath();
28      lienzo.strokeStyle="rgb(0,255,0)";
29      lienzo.lineCap="butt";
30      lienzo.moveTo(30,5);
31      lienzo.lineTo(30,295);
32      lienzo.stroke();
33      lienzo.strokeStyle="rgb(212,51,255)";
34      lienzo.lineCap="round";
35      lienzo.beginPath();
36      lienzo.moveTo(50,5);
37      lienzo.lineTo(50,295);
38      lienzo.stroke();
39      lienzo.strokeStyle="rgb(125,60,152)";
40      lienzo.lineCap="square";
41      lienzo.beginPath();
42      lienzo.moveTo(70,5);
43      lienzo.lineTo(70,295);
44      lienzo.stroke();
45    }
```





```
46   lienzo.beginPath();
47   lienzo.strokeStyle="rgb(236,255,51)";
48   lienzo.lineJoin="bevel";
49   lienzo.moveTo(100,90);
50   lienzo.lineTo(130,10);
51   lienzo.lineTo(160,90);
52   lienzo.stroke();
53
54   lienzo.beginPath();
55   lienzo.lineJoin="round";
56   lienzo.moveTo(100,180);
57   lienzo.lineTo(130,100);
58   lienzo.lineTo(160,180);
59   lienzo.stroke();
60
61   lienzo.beginPath();
62   lienzo.lineJoin="miter";
63   lienzo.moveTo(100,270);
64   lienzo.lineTo(130,190);
65   lienzo.lineTo(160,270);
66   lienzo.stroke();
67
68 }
69 }
70 </script>
71
72 </head>
73 <body onLoad="dibujar()">
74 <canvas id="lienzo1" width="300" height="300">
75 Su navegador no permite utilizar canvas.
76 </canvas>
77 </body>
78 </html>
```

4. Corregir errores.
5. Abrir el explorador de archivos.
6. Guardar el archivo.
7. Ejecutar la página, se deberá visualizar como la siguiente imagen.



**Retroalimentación**

1. Con qué instrucción se denota el espacio vectorial en HTML5
2. Para dibujar un vector se anota la instrucción lienzo.LineTo(30,295), anota las partes de la instrucción.
3. Anotar las instrucción para que la punta de las líneas sean redondas
4. Anotar la instrucción para que las líneas amarillas sean de color negro
5. Qué hace la instrucción lienzo.lineCap="square";
6. Crear la siguiente imagen, con líneas de diferentes colores





## Unidad de competencia II.

Aplicar transformaciones básicas elementales en 2D y transformaciones de encuadre y perspectiva

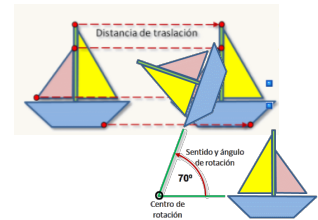
### Práctica 2: Traslación, Escalado, Rotación y Reflexión.

#### Introducción

**Traslación:** es el movimiento directo de una figura en la que todos sus puntos: Se mueven en la misma dirección y en la misma distancia. El resultado de una traslación es otra figura idéntica que se ha desplazado una distancia en una dirección determinada (González et all, 2012).

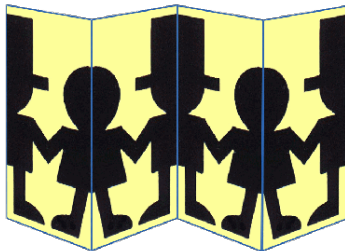
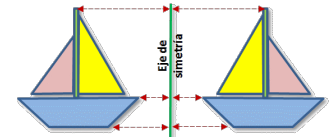
**Rotación o giro:** es un movimiento alrededor de un punto que mantiene la forma y el tamaño de la figura original. Una rotación se determina por estos tres elementos (López, 2014)

- Un **ángulo** que determina la amplitud de la rotación.
- Un punto llamado **centro** de rotación.
- Un **sentido** de la rotación que puede ser del mismo sentido de las agujas del reloj o en sentido contrario.



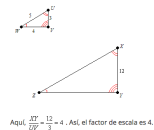
#### Simetría:

La simetría respecto a un eje es una reflexión y se caracteriza porque: Los puntos simétricos de una figura y los de la figura reflejada están sobre la **misma línea**, los puntos de ambas figuras están a la **misma distancia** del eje de simetría en direcciones opuestas y La figura reflejada siempre tiene el mismo tamaño, pero en la **dirección opuesta**. (Costa, 2015).



#### Escalado

Es la representación dibujada en un mapa, carta náutica o un plano con escala unidad por unidad, donde cada segmento muestra la relación entre la longitud de la representación y la de la realidad.



#### Descripción de la práctica

Crear dibujos 2d en HTML5 que representen la Traslación, Escalado, Rotación y Reflexión. Estos dibujos se realizan a través de código en JavaScript.

**Duración:** 1 horas

**Objetivos de aprendizaje:** El alumno aplicará las transformaciones básicas elementales en 2D y de perspectiva como traslación, escalado, rotación y reflexión.

#### Requisitos Previos

Saber los conceptos de traslación, escalado, rotación y reflexión. Saber programar en HTML5





### Materiales

Computadora

Navegador

Editor de texto (puede ser sublime o Notepad++)

### Procedimiento

1. Crear una carpeta con el nombre práctica2
2. Copiar una imagen en la carpeta en donde se ubicará el script html
3. Ejecutar Sublime o Notepad++
4. Crear un archivo con extensión HTML
5. Copiar el siguiente código

```
<!DOCTYPE HTML>
<html>
<head>

<script type="text/javascript">
function retornarLienzo(x) {
  var canvas = document.getElementById(x);
  if (canvas.getContext) {
    var lienzo = canvas.getContext("2d");
    return lienzo;
  } else
    return false;
}

function dibujar() {
  var lienzo=retornarLienzo("lienzo1");
  if (lienzo) {
    lienzo.clearRect(0,0,600,600);
    lienzo.save();
    lienzo.translate(300,300);
    lienzo.rotate(avance);
    lienzo.scale(tamx,tamy);
    lienzo.drawImage(img1,-125,-125);
    avance+=0.05;
    tamx+=0.01;
    tamy+=0.01;
    if (avance>Math.PI*2)
      avance=0;
    if (tamx>=10) {
      tamx=0.01;
      tamy=0.01;
    }
    lienzo.restore();
  }
}

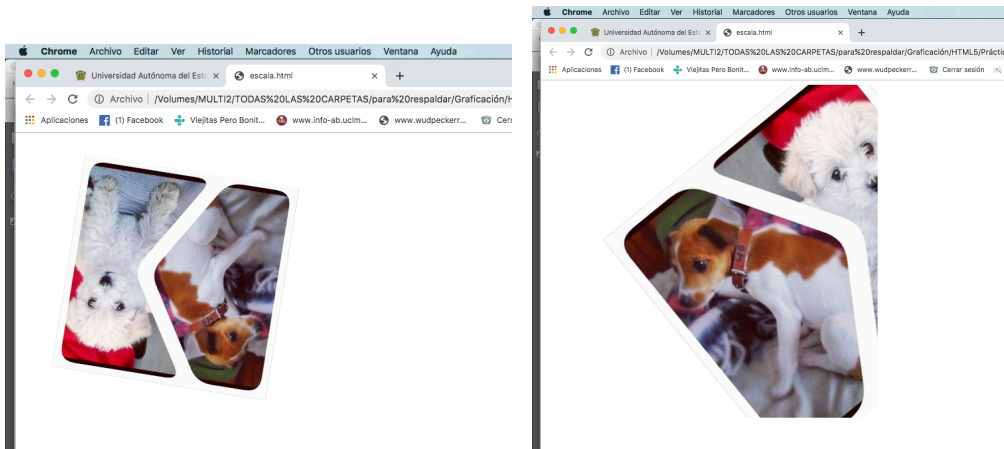
var avance=0;
var img1;
var tamx=0.01;
var tamy=0.01;
function inicio() {
```



```
img1 = new Image();  
img1.src = 'perros.png';  
img1.onload = function(){  
    setInterval(dibujar,50);  
}  
}  
  
</script>  
  
</head>  
<body onLoad="inicio()">  
<canvas id="lienzo1" width="600" height="600">  
Su navegador no permite utilizar canvas.  
</canvas>  
</body>  
</html>
```

## 6. Ejecutar el archivo

La imagen que se verá en el navegador deberá ser similar a la siguiente:



## Retroalimentación

1. Anota las instrucciones en HTML5 para escalar, rotar, traslación y reflexión.
2. Cambia la instrucción `lienzo.translate(300,300);` por `lienzo.translate(600,300);` anota que es lo que cambió
3. Cambia el tamaño máximo al que debe llegar del escalado por 2 y anota cuantas veces cambió la imagen, después por 3 y anota cuantas veces cambió la imagen y al final por 5 y anota cuantas veces cambió la imagen. Anota tus comentarios en relación a la instrucción del escalado.

## Unidad de competencia III.

Modelar objetos

### Práctica 3: Crear modelos poligonales y poliédricos.

#### Introducción

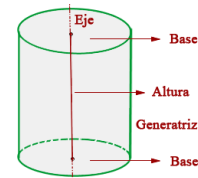
¿Qué es un polígono?

Por definición, en el modelado tridimensional, es una forma bidimensional compuesta de líneas rectas y conectadas –y consecuentemente, cerrada–. A la hora de modelar personajes para videojuegos, el tema que nos ocupa, la forma más común es el triángulo, por lo que la inmensa mayoría de personajes que vemos en los títulos "actuales" están realizados de polígonos triangulares (Palmer et al, 2013).

A mayor potencia de la máquina, mayor cantidad de polígonos que puede mover en pantalla. Proporcionalmente, a mayor número de polígonos, mayor será la cantidad de detalle de un elemento en concreto. Hoy nos centraremos en los modelados poligonales de los personajes, pero, obviamente, esto que comentamos también afecta a los escenarios. Claro que, según evolucionan las técnicas de desarrollo, también es posible añadir detalles de otras maneras que no implican sumar triángulos. Pero no adelantemos detalles.

**Cilindro.** Cuerpo engendrado por un rectángulo que gira alrededor de uno de sus lados

- Eje: lado fijo alrededor del cual gira el rectángulo
- Generatriz: lado opuesto al eje. Engendra al cilindro
- Bases: círculos que conforman los lados perpendiculares al eje
- Altura: distancia entre las dos bases



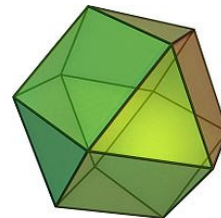
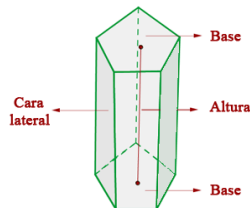
**Prisma.** Sólido determinado por dos polígonos paralelos y congruentes que se denominan bases. Sus caras laterales son paralelogramos.

Elementos de un prisma

#### Poliedro

Sólido determinado por polígonos

- Caras: polígonos que limitan al poliedro
- Aristas: lados de las caras del poliedro
- Vértices: puntos concurrentes de las aristas



#### Descripción de la práctica

Crear modelos poligonales y poliédricos en un programa de modelado para imágenes 3D. En este caso se utiliza SketchUp en línea (<https://www.sketchup.com/es/plans-and-pricing/sketchup-free>)

**Duración:** 2 horas





**Objetivos de aprendizaje:** El alumno aprenderá y aplicará las técnicas de modelado tridimensional para la creación de modelos poligonales y poliédricos.

### Requisitos Previos

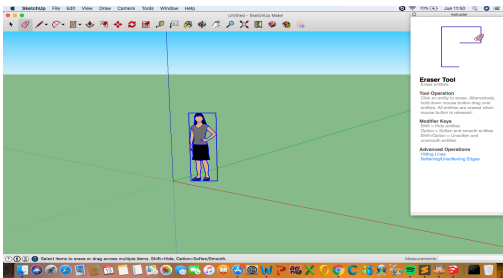
Tener conocimientos en sketchUp y conceptos sobre los modelos 3D

### Materiales

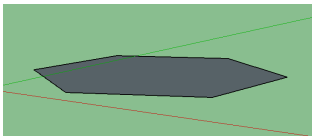
Computadora  
SketchUp

### Procedimiento

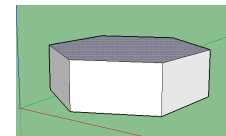
- 1.- Abrir SketchUp
2. Eliminar la muñeca, haciendo clic en la goma que se encuentra en la barra de herramientas




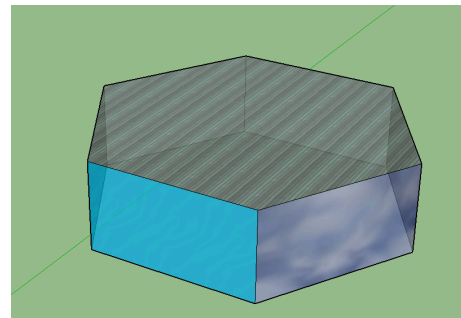
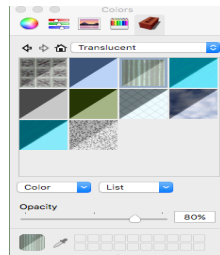
3. Crear un polígono con la herramienta



4. Elegir la herramienta empujar  y estirar y estirar hacia arriba



5. Pintar el modelo con la herramienta  eligiendo transparencias para que se noten todos sus lados-



### Retroalimentación

1. Que herramientas usarías para crear un prisma
2. Que herramientas usarías para crear un poliedro de 8 lados
3. Crear un cubo y pintarlo de materiales
4. Modificar el cubo y crear un robot







## Unidad de competencia IV.

Modelar objetos en 2D y 3D.

### Práctica 4: Modelar objetos 2d

#### Introducción

Un **modelo geométrico 2D** es un modelo geométrico de un objeto como figura bidimensional, por lo general sobre un plano euclidiano o cartesiano.

Aunque todos los objetos materiales son tridimensionales, un modelo geométrico 2D es a menudo suficiente para ciertos objetos planos, tales como recortes de papel.

Los modelos geométricos 2D son también convenientes para describir ciertos tipos de imágenes artificiales, tales como diagramas técnicos, logotipos, glifos de una fuente, etc. Son una herramienta esencial de la infografía y a menudo utilizados como componentes de los modelos geométricos en 3D.

Un dibujo 2d puede contener diversos dibujos que unidos muestran una imagen. Un polígono es la unión de varios vectores almacenados en un arreglo. Para dibujar polígonos en JavaScript se utiliza la instrucción `Lienzo.beginPath`.

#### Descripción

La práctica crea modelos 2d mediante scripts realizados en HTML5, en donde se agregan vectores, colores, funciones de tipo aleatoria y letras, todo para crear un barco con el nombre del alumno.

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno generará imágenes en 2D para que al final se vea una imagen (barco).

#### Requisitos Previos

Tener conocimientos sobre HTML5 y dibujo de vectores en 2D.

#### Materiales

Computadora

Sublime

navegador

#### Procedimiento

1. Ejecutar Sublime o Notepad++
2. Crear un archivo con extensión HTML
3. Copiar el siguiente código

```
<html>
<!DOCTYPE HTML>
<head>
<script type="text/javascript">

function retornarLienzo(x)
{
    var canvas = document.getElementById(x);
    if(canvas.getContext)
    {
        var lienzo = canvas.getContext("2d");
        return lienzo;
    }
    else
```





```
return false;
}

var fila=0;

function dibujar()
{
    var lienzo=retornarLienzo("lienzo1");
    if(lienzo){
        lienzo.strokeStyle = "rgb(0,0,0)";
        lienzo.lineWidth = 10;
        lienzo.strokeRect(10,10,400,400);
        lienzo.fillStyle="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*
255)+" "+Math.round(Math.random()*255)+"")";
        lienzo.fillRect(100,265,200,85);
        lienzo.fill();
        lienzo.fillStyle="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*
255)+" "+Math.round(Math.random()*255)+"")";
        lienzo.beginPath();
        lienzo.moveTo(210,30);
        lienzo.lineTo(210,260);
        lienzo.lineTo(360,210);
        lienzo.fill();
        lienzo.fillStyle="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*
255)+" "+Math.round(Math.random()*255)+"")";
        lienzo.beginPath();
        lienzo.moveTo(100,200);
        lienzo.lineTo(200,260);
        lienzo.lineTo(200,13);
        lienzo.fill();

        lienzo.fillStyle="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*
255)+" "+Math.round(Math.random()*255)+"")";
        lienzo.beginPath();
        lienzo.moveTo(25,265);
        lienzo.lineTo(95,265);
        lienzo.lineTo(95,350);
        lienzo.fill();

        lienzo.fillStyle="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*
255)+" "+Math.round(Math.random()*255)+"")";
        lienzo.beginPath();
        lienzo.moveTo(305,265);
        lienzo.lineTo(380,265);
        lienzo.lineTo(305,350);
        lienzo.fill();

        lienzo.fillStyle
="rgb("+Math.round(Math.random()*255)+" "+Math.round(Math.random()*255)+" "+Math.round(
Math.random()*255)+"")";
        lienzo.beginPath();
        lienzo.arc(150,265+42.5,20,0,Math.PI*2,true);
        lienzo.closePath();
        lienzo.fill();
    }
}
```



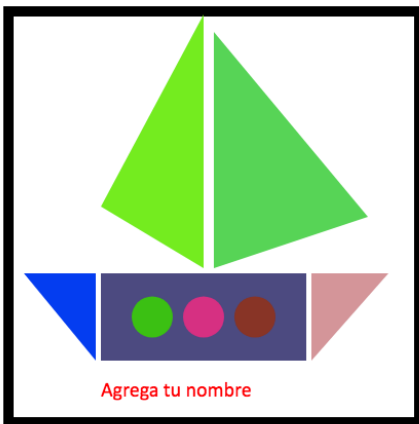


```
        lienzo.fillStyle
        ="rgb("+Math.round(Math.random()*255)+","+Math.round(Math.random()*255)+","+Math.round(
        Math.random()*255)+)";
        lienzo.beginPath();
        lienzo.arc(200,265+42.5,20,0,Math.PI*2,true);
        lienzo.closePath();
        lienzo.fill();

        lienzo.fillStyle
        ="rgb("+Math.round(Math.random()*255)+","+Math.round(Math.random()*255)+","+Math.round(
        Math.random()*255)+)";
        lienzo.beginPath();
        lienzo.arc(250,265+42.5,20,0,Math.PI*2,true);
        lienzo.closePath();
        lienzo.fill();

        lienzo.font = '15pt Calibri';
        lienzo.fillStyle = 'red';
        lienzo.fillText('Agrega tu nombre', 100,385);
    }
</script>
</head>
<body onLoad="dibujar()">
<canvas id="lienzo1"width="1000"height="6000">
    Su navegador no permite utilizar canvas.
</canvas>
</body>
</html>
```

4. Depurar los errores.
5. Abrir el navegador y ejecutar la página, se deberá visualizar como a continuación se muestra:



6. Verificar que cada vez que se actualice la página cambie de colores y agregue tu nombre.

**Retroalimentación**

1. Anotar las instrucciones que se utilizaron para las velas.
2. Anotar las instrucciones que se utilizaron para los círculos.
3. Anotar las instrucciones que se utilizaron para las letras
4. Anotar la instrucción que se utilizó para cambiar de colores aleatoriamente.
5. La imagen se dibujó en un plano cartesiano x y y? O x,y y z?





## Unidad de competencia IV.

Modelar objetos en 2D y 3D.

### Práctica 4: Modelar objetos 3d

#### Introducción

El **modelado 3D** es el proceso de desarrollo de una representación matemática de cualquier objeto de tres dimensiones (ya sea inanimado o vivo) a través de un software especializado (Jocquera, 2018). Al producto se le llama **modelo 3D** y puede visualizarse como una imagen de 2 dimensiones de fenómenos físicos. El modelo también se puede crear físicamente usando dispositivos de impresión 3D.

Los modelos pueden ser creados automáticamente o manualmente. El proceso manual de preparar la información geométrica para los gráficos 3D es similar al de las artes plásticas y la escultura.

El *software de modelado 3D* es un tipo de software utilizado para producir modelos tridimensionales. Los programas individuales de este tipo son llamados «Aplicaciones de modelado» o «modeladores». En este caso se utilizará

#### Descripción

La práctica modela una casa con escalones y chimenea, utilizando diversas herramientas para que el alumno aprenda sobre ejes, vectores, modelos, colores.

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno generará imágenes en 3D (casa).

#### Requisitos Previos

Tener conocimientos sobre modelos 3D y SketchUp

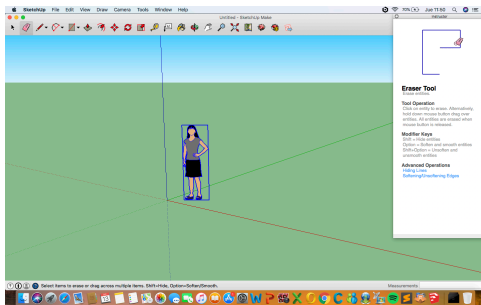
#### Materiales

Computadora  
SketchUp

#### PROCEDIMIENTO

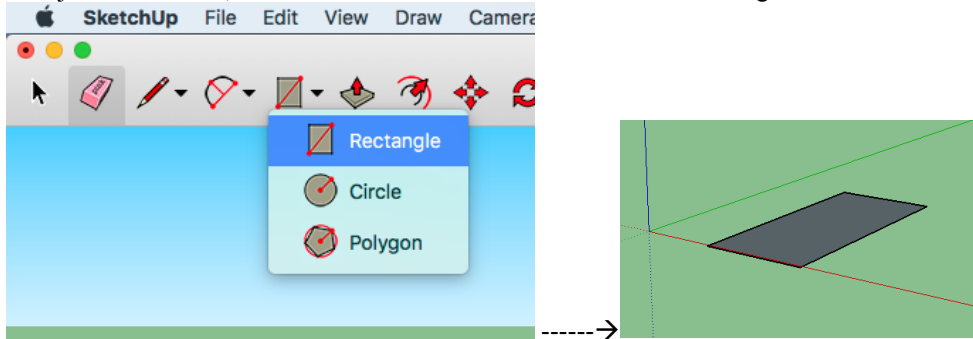
1. Abrir SketchUp

Eliminar la muñeca, haciendo clic en la goma que se encuentra en la barra de herramientas

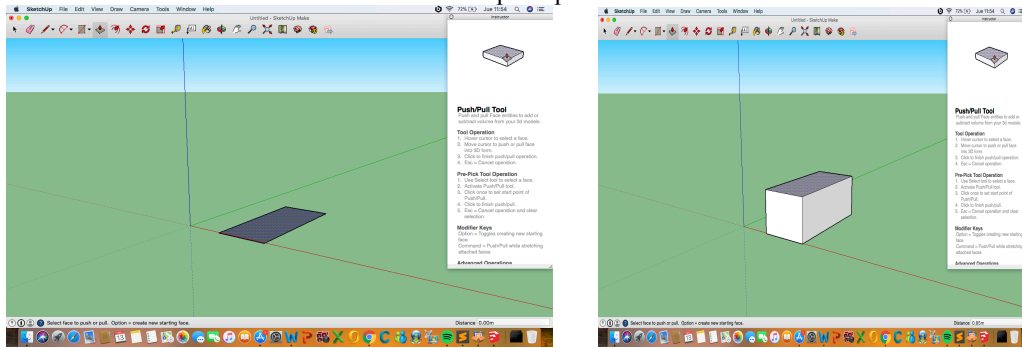




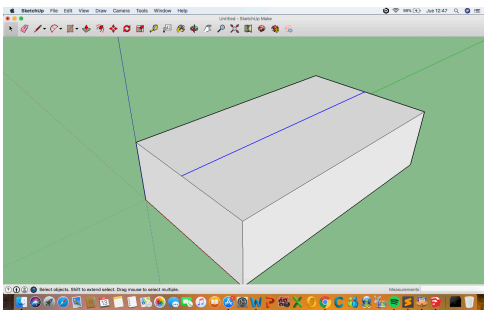
2. Dibujar un cuadrado, haciendo clic en la barra de herramientas Rectangle



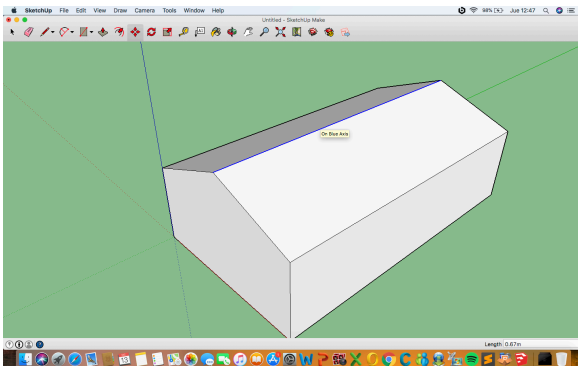
3. Crear el cubo haciendo clic en la herramienta push/pull



4. Dibujar una línea con la herramienta lápiz, haciendo clic en el punto medio y finalizando en el punto medio.



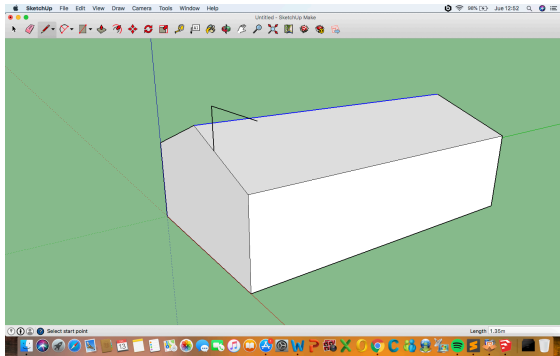
5. Seleccionar la línea con la herramienta de selección y subirla para simular un techo.



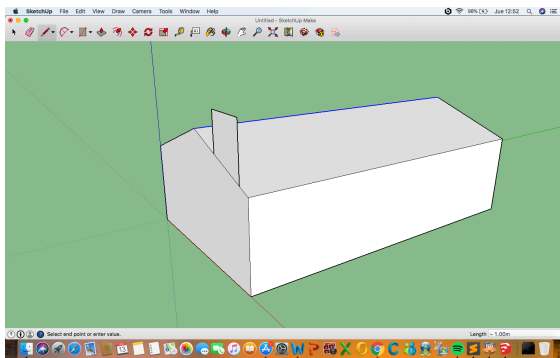


**Hacer la chimenea:**

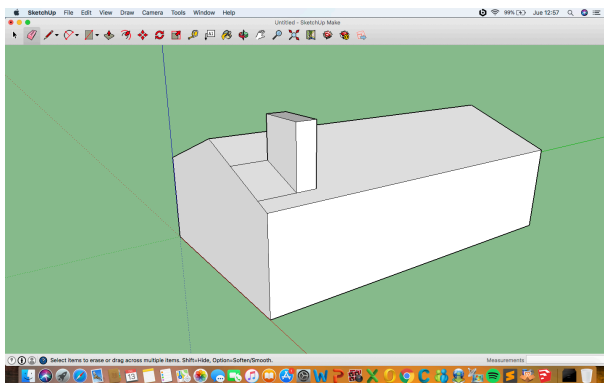
6. hacer una línea hacia arriba, cuidar que vaya en el eje azul



7. Completar un cuadro cuidando que vaya sobre el eje rojo para que vaya perpendicular al piso.



8. Empujar para hacer el cubo de la chimenea con la herramienta push/pull.



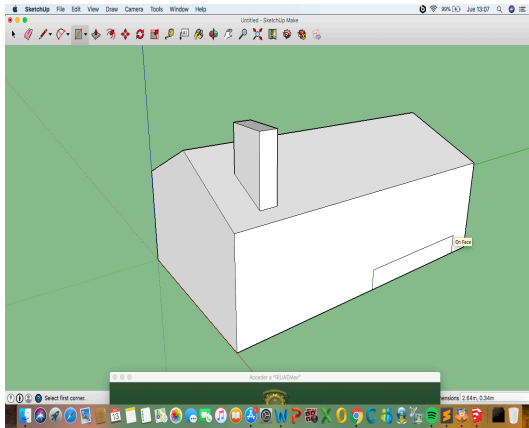
9. Borrar las líneas sobrantes con la goma.



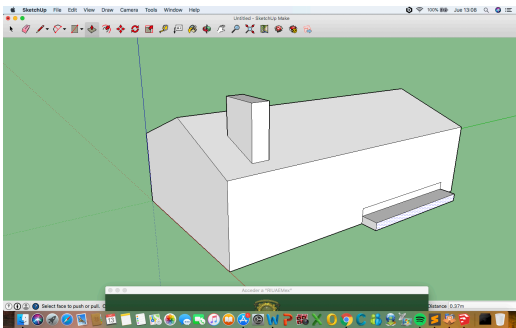



Ahora crear unas escaleras:

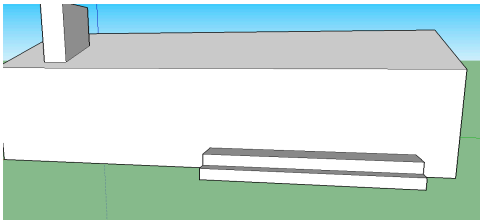
10. Dibujar un rectángulo.



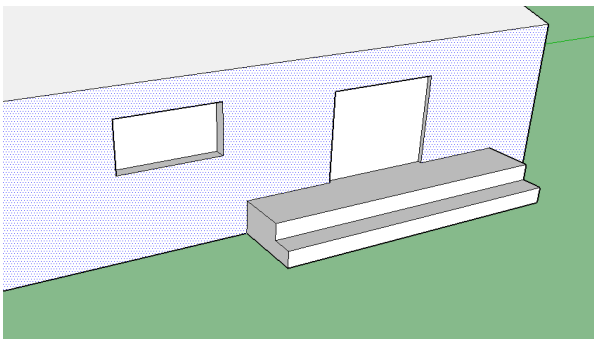
11. Dividirlo por la mitad



12. Empujar las mitades para tener 2 escalones con la herramienta push/pull 

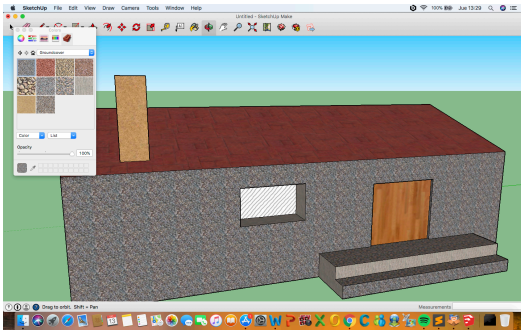


13. Ahora dibujar la puerta y la ventana con la herramienta cuadrado y push 





16. Pintar la casa con la herramienta pintura, no olvides pintarla toda.



### Retroalimentación

1. Anota las herramientas que utilizaste para realizar la imagen.
2. Pinta de otros colores la casa.
3. Que ejes intervienen en el desarrollo de esta imagen.
4. Pinta la ventana de tal manera que parezca de vidrio transparente.
5. Agrega un sol en la parte del cielo.







## Unidad de competencia V.

Visualizar objetos de una escena en la pantalla controlando las condiciones lumínicas

### Práctica 5: Luces

#### Introducción

Puede conseguirse un mayor realismo en las imágenes si se representan las superficies de los objetos utilizando las condiciones de iluminación de una escena y las características asignadas a las superficies.

Las condiciones de iluminación se establecen especificando el color y la ubicación de las fuentes de luz, y también pueden definirse efectos de iluminación de fondo.

Las propiedades de las superficies de los objetos incluyen, por ejemplo, información sobre si una superficie es transparente u opaca, o si es suave o rugosa. Podemos asignar valores a una serie de parámetros para modelar superficies tales como el cristal, el plástico, superficies de madera veteada o incluso la apariencia rugosa de una naranja.

La teoría del color es un grupo de reglas básicas en la mezcla de colores para conseguir el efecto deseado combinándolos. El color es una sensación producida por el reflejo de la luz en la materia y transmitida por el ojo al cerebro (López. 2014).

La mezcla de los colores primarios de la luz, que son rojo, verde y azul (RGB, iniciales en inglés de los colores primarios), se realiza utilizando el sistema de color aditivo, también conocido como el modelo RGB o el espacio de color RGB. ... Cuando ningún color luz está presente, se percibe el negro.

La luz blanca se puede producir combinando el rojo, el verde y el azul, mientras que combinando pigmentos cian, magenta y amarillo se produce el color negro.

#### Descripción

La práctica realiza una casa en donde se agregan imágenes para simular luces dentro de la casa. Existen programas en donde las luces ya están como recursos adicionales, sin embargo con este procedimiento es una manera fácil de agregar luces a su modelos. Además, la combinación de colores le dan realce.

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno generará imágenes en 3D con simulación a luces

#### Requisitos Previos

Tener conocimientos sobre el programa SketchUP


#### Materiales

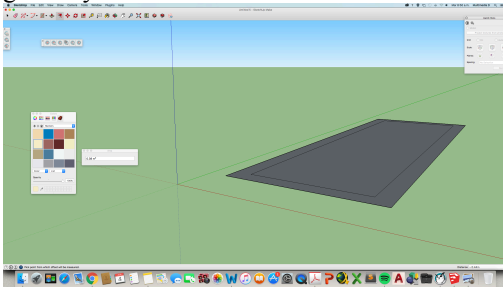
Computadora  
SketchUP




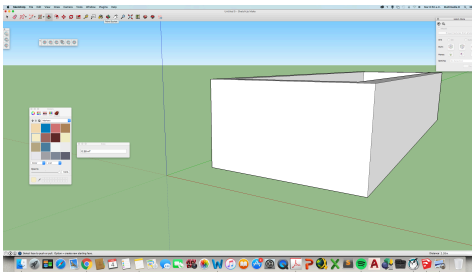


### Procedimiento

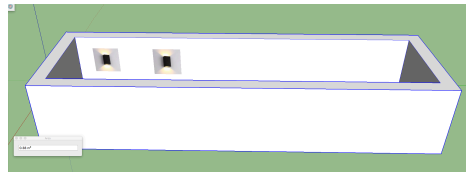
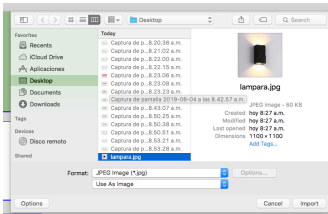
1.- Dibujar un rectángulo con un rectángulo dentro. Esto se hace con la herramienta rectángulo para el más grande y con la herramienta offset 



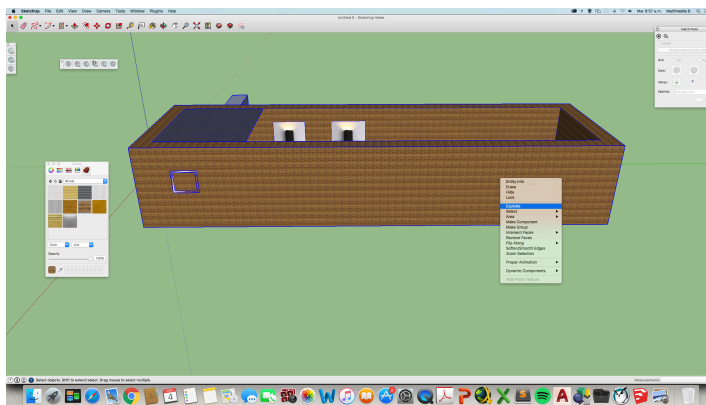
2. Estirar el rectángulo de afuera con la herramienta push/pull 



3. Importar lámparas como imagen y colocarlas en la pared, esto simulará que la pared tiene lámparas.



4. Pintar la casa para contrastar y agregar un techo para que puedas observar las luces por parte de adentro de la casa.





**Actividades de Retroalimentación**

1. Qué tipo de luz se determina cuando ningún color está presente?
2. Qué colores primarios se combinan para hacer todos los colores?
3. Cuales son los rangos de números que combinan los colores?
4. Aplica nuevos colores a la imagen elaborada para hacerla parecer con sobras.
5. Aplica nuevos colores a la imagen elaborada para hacerla parecer con luz.





## Unidad de competencia VI.

Generar imágenes por computador que emulen a la realidad y animar los objetos presentes en la escena añadiéndoles movimiento.

### Práctica 6: Animación 3D

#### Introducción

El **modelado 3D** es el proceso de desarrollo de una representación matemática de cualquier objeto de tres dimensiones (ya sea inanimado o vivo) a través de un software especializado. Al producto se le llama **modelo 3D**. Se puede visualizar como una imagen de 2 dimensiones de fenómenos físicos. El modelo también se puede crear físicamente usando dispositivos de impresión 3D.

Los modelos pueden ser creados automática o manualmente. El proceso manual de preparar la información geométrica para los gráficos 3D es similar al de las artes plásticas y la escultura.

El *software de modelado 3D* es un tipo de software utilizado para producir modelos tridimensionales. Los programas individuales de este tipo son llamados «Aplicaciones de modelado» o «modeladores».

#### Descripción

La práctica dibuja un modelo 3d de una lata para luego agregar movimiento a los objetos mediante el movimiento de la cámara, esto simula una escena animada. Se utiliza el modelo de la lata aunque de puede aplicar a cualquier modelo, esto para prever que el alumno tenga el modelo 3d y pueda aplicar el movimiento.

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno generará imágenes en 3D y les aplicará movimiento para crear una escena animada.

#### Requisitos Previos

Tener conocimientos sobre de SketchUP

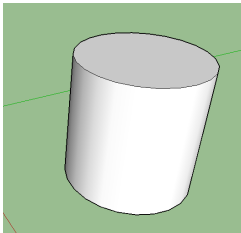
#### Materiales

Computadora

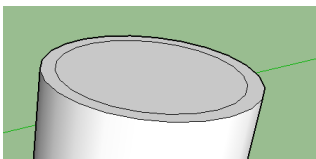
SketchUp

Hacer el modelo 3d de una lata.

1. Crear un cilindro

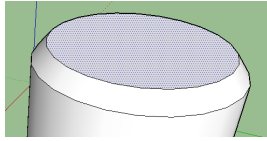



2. Crear un circo concéntrico, con la herramienta offset

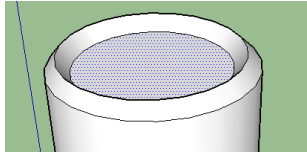




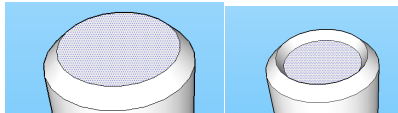
3. Dar forma con la herramienta de mover y la tela alt



4. Hacer otro círculo en el centro con offset  y empujar hacia dentro. Tener cuidado de primero seleccionarlo y después empujar.

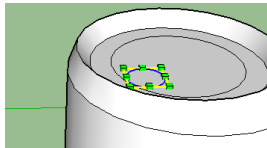


5. Repetimos el procedimiento en la parte de abajo

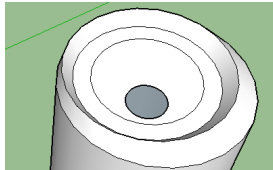


6. Importar una imagen que se asemeje a la tapa de una lata

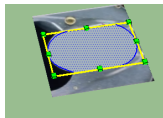
7. Acercar la cámara y dibujar con el arco un círculo fuera de la lata y colocarlo sobre la lata



8. Seleccionar el círculo y suprimir para generar un hoyo

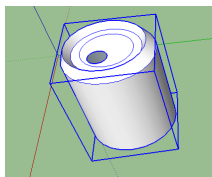



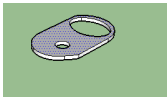
9. Dibujar con curvas y lápiz el corcho y escalarlo si es necesario



10. Hacer lo mismo con las demás componentes

11. Seleccionar la lata y agruparla, dando clic en edit → Make Group



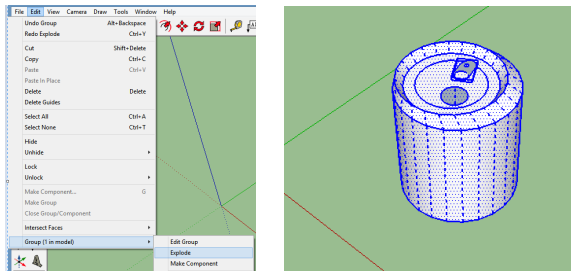
12. Dar volumen a la figura  -> 

13. Voltar la lata con 



14. Seleccionar la figura y colocarla en la parte superior de la lata

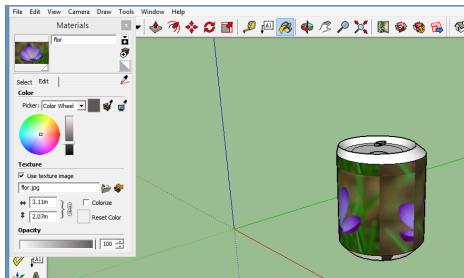
15. Aplicar el explode a la lata



16. Importar la imagen, activando la casilla “usar como textura” (Para la textura deberás tomarte una foto y utilizarla como textura para tu lata)

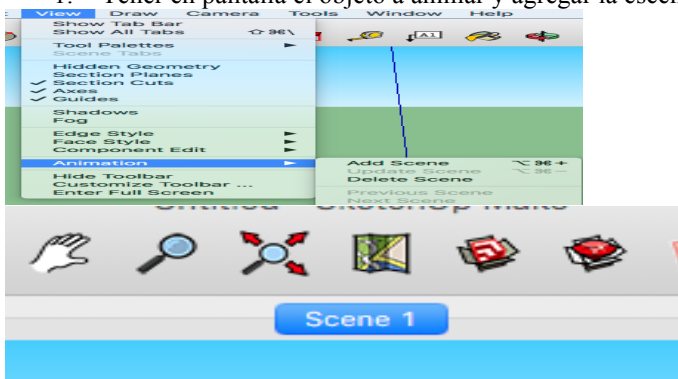
17. Lo colocamos de punto a punto

17. Seleccionamos el tarro de pintura, seleccionamos la imagen que importamos y aplicamos el bote pintura



## Crear la Animación

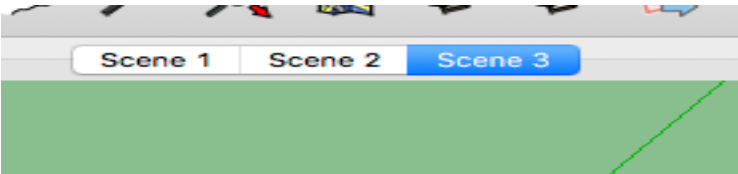
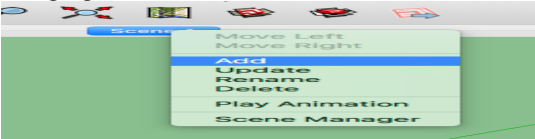
1. Tener en pantalla el objeto a animar y agregar la escena



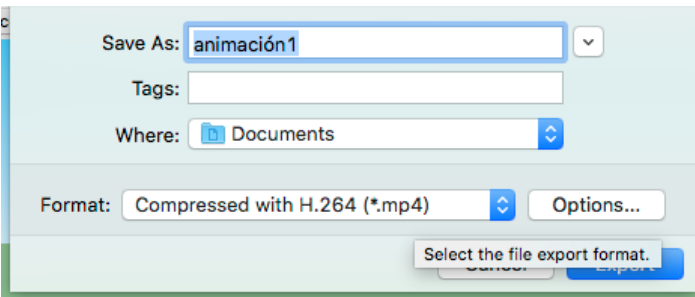
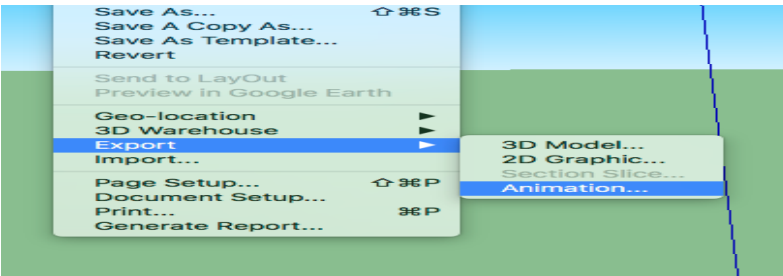
2. Mover los objetos con la herramienta orbita



### 3. Agregar más escenas



### 4. exportar como video



### Preguntas de Repaso

1. Se le llama animación \_\_\_\_\_ a la que se genera la secuencia de imágenes por métodos pictóricos, lo que entendemos por una imagen “dibujada”, formada por pixeles cuya coloración se asigna manualmente o semiautomáticamente, por mecanismos sencillos guiados de forma manual.  
a) Animación clásica      b) Animación de síntesis por ordenador      c) Animación fotograma a fotograma
2. Se le llama animación \_\_\_\_\_ a la que se crea las imágenes por un proceso automático a partir de una representación de los objetos que forman parte de la escena y de su movimiento.  
a) Animación clásica      b) Animación interactiva      c) Animación fotograma a fotograma





Si los fotogramas se van generando a partir de una especificación previa y se van almacenando en ficheros (o en un fichero único para toda la animación) o en cualquier otro soporte analógico o digital, hablaremos de una

a) Animación clásica      b) Animación de síntesis por ordenador      c) Animación fotograma a fotograma

3. se produce cuando el proceso que crea la animación va mostrando los fotogramas inmediatamente después de producirlos, permitiendo al usuario responder inmediatamente y teniendo en cuenta estas respuestas en la síntesis de imágenes.

a) animación interactiva      b) Animación en tiempo real      c) Animación fotograma a fotograma

4. Cuando las restricciones temporales de la animación interactiva son relativamente estrictas se habla de:

a) animación interactiva      b) Animación en tiempo real      c) Animación fotograma a fotograma







## Referencias

- Costa G. A. (2015). Una introducción a la simetría, Educación permanente, Madrid 2015.
- González M. C., Albusac J.J., Pérez C.S., López G. J., Mora C.C(2012). Desarrollo de videojuegos, programación gráfica. Universidad de Alicante, España.
- Jocquera O. A.(2018). Fabricación digital: Introducción al modelado e impresión 3d., CamSda, Madrid España.
- López, L.A.(2014). Diseño Gráfico, fundamentos y técnicas, Multimedia Anaya, Madrid España.
- Mejía D.F., Álvarez J. R. Y Fernández C., H. (2005). Matemáticas previas al cálculo. Universidad de Medellín, Colombia.
- Navarro, L. J.(2009). Álgebra lineal, Espacio Vectorial, Espacio Vectorial Euclídeo, Universidad Alicante, España.
- Palmer, I.C., Bibb S.F., Jarvis, J.A., Mrachek, L.A(2013). Matemáticas prácticas, aritmética, algebra, trigonometría y regla de cálculo, Editorial Reverté, S.A, Barcelona, España.

