

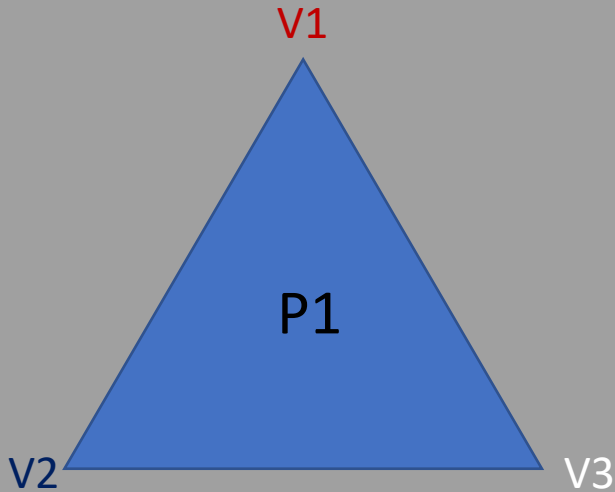
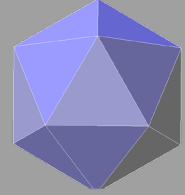
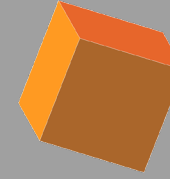
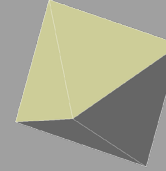
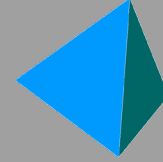


Representación tridimensional de objetos



MODELO POLIÉDRICO

Este es un modelo de representación de objetos en 3D que consiste en formar una superficie con la unión de polígonos.



Una superficie puede definirse con dos matrices, una que contiene todos los vértices y otra que hace referencia a los vértices que conforma cada polígono independiente.

	<i>Vertices</i>				<i>Polígonos</i>		
<i>V1:</i>	<i>X1</i>	<i>Y1</i>	<i>Z1</i>	<i>P1:</i>	<i>V1</i>	<i>V2</i>	<i>V3</i>
<i>V2:</i>	<i>X2</i>	<i>Y2</i>	<i>Z2</i>	<i>P2:</i>
<i>V3:</i>	<i>X3</i>	<i>Y3</i>	<i>Z3</i>	<i>P3:</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<i>Vn:</i>	<i>Xn</i>	<i>Yn</i>	<i>Zn</i>	<i>Pm:</i>

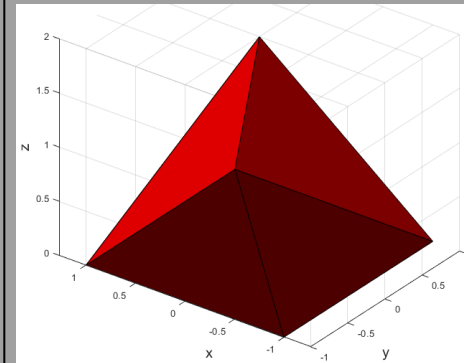
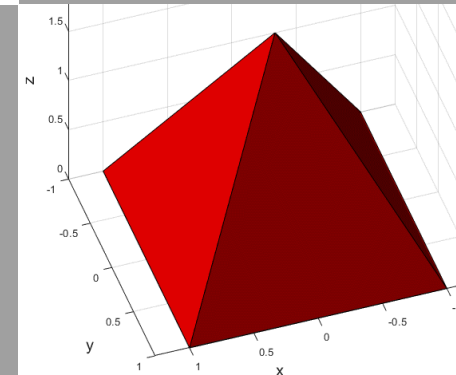
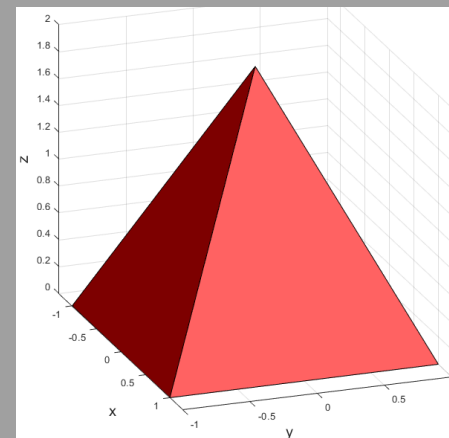


EJEMPLO DE PIRÁMIDE - MATLAB

La función `patch()` dibuja la pirámide.

Código 1.

```
figure(1)
clf;
hold on
grid on
axis equal;
light;
view(70,20);
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
zlabel('z','FontSize',15);
color = [1,0,0];
color_borde = [0,0,0];
vertices = [ 0, 0,2; % Esta es la matriz de vertices
            1, 1,0;
            -1, 1,0;
            -1,-1,0;
            1,-1,0];
caras = [1,2,3; % Esta es la matriz de polígonos
        1,3,4; % Hace referencia a los índices
        1,4,5; % de la matriz vertices
        1,5,2;
        2,4,3;
        2,4,5];
piramide = patch('Faces',caras,'Vertices',vertices,'FaceColor',
color,'EdgeColor',color_borde);
```

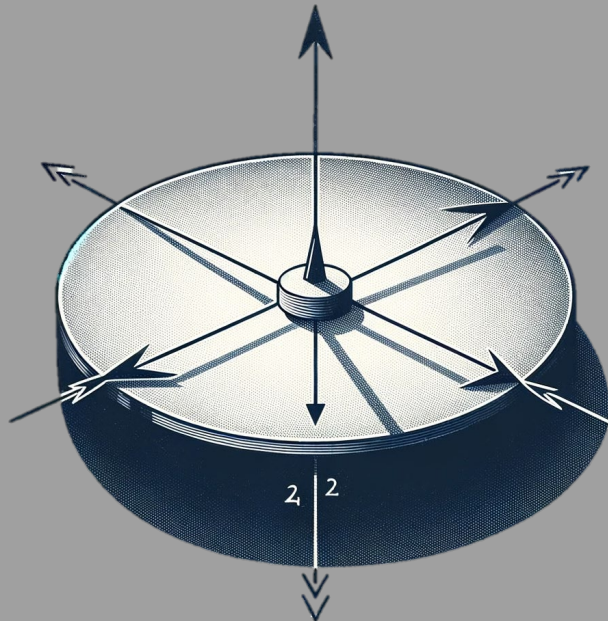


Taller

1.- Diseñar un algoritmo que calcule un vector unitario (v_p) el cual es perpendicular a un vector dado ($V_i = v_{ix}i_x + v_{iy}j_x + v_{iz}k_x$). Debe verificarse la siguiente propiedad:

$$v_i \cdot v_p = 0$$

Nota: Obsérvese que existen infinitos vectores que cumplen esta propiedad. Por lo tanto se debe seleccionar alguno de ellos.



Taller

2. Crear una función que dibuje un cono, tomando como datos de entrada un vector que indica la posición del cono (centro de la base del cono), un vector que indica la dirección del cono (orientación), un escalar que indica el diámetro de la base del cono, un vector que indica el color de la superficie del cono (color = [R,G,B]) y un vector que indica el color de los bordes del cono.

Ejemplo:

Código 2.

```
clear;  
clc;  
pos = [0,0,10];  
vec = [1,1,1];  
t = 5;  
d = 4;  
color = [1,0,0];  
color_borde = [0,0,0];  
figure(1)  
clf;  
hold on
```

```
grid on  
axis equal;  
light;  
view(70,20);  
xlabel('x','FontSize',15);  
ylabel('y','FontSize',15);  
zlabel('z','FontSize',15);  
dibujar_cono(pos, vec,t,d,  
color, color_borde);
```



Taller

Procedimiento:

- Dado el vector que indica la dirección del cono (vec), hallar un vector unitario perpendicular al mismo. Haciendo uso de este vector unitario y conociendo el diámetro de la base del cono, calcular un punto ubicado sobre la base de cono.
- A partir del punto calculado, el vector que indica la dirección y el diámetro del cono, obtener n puntos ubicados sobre la base del cono. Tomar $n = 10$ y realizar una grafica similar a la figura 1.

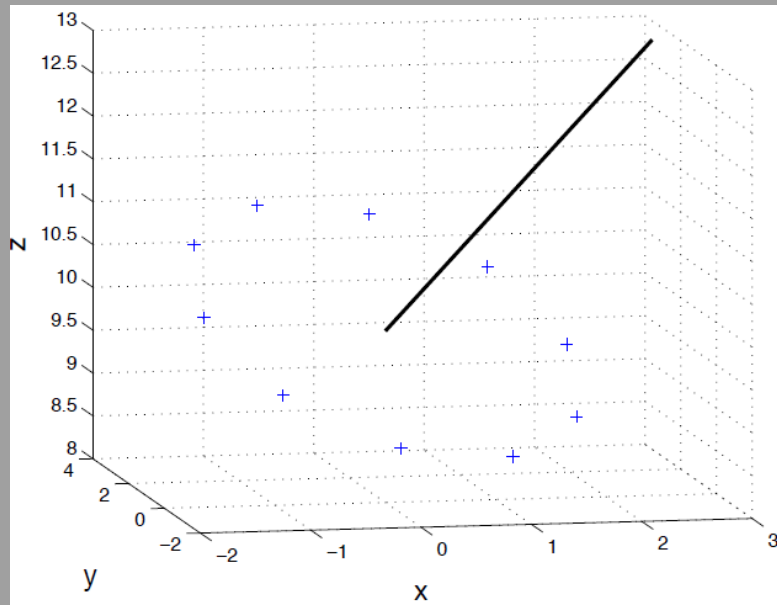


figura 1. Generación de n puntos ubicados sobre la base del cono

Taller

- c) Dibujar un triangulo que tenga como vértices dos puntos consecutivos de la base del cono y la punta del mismo.

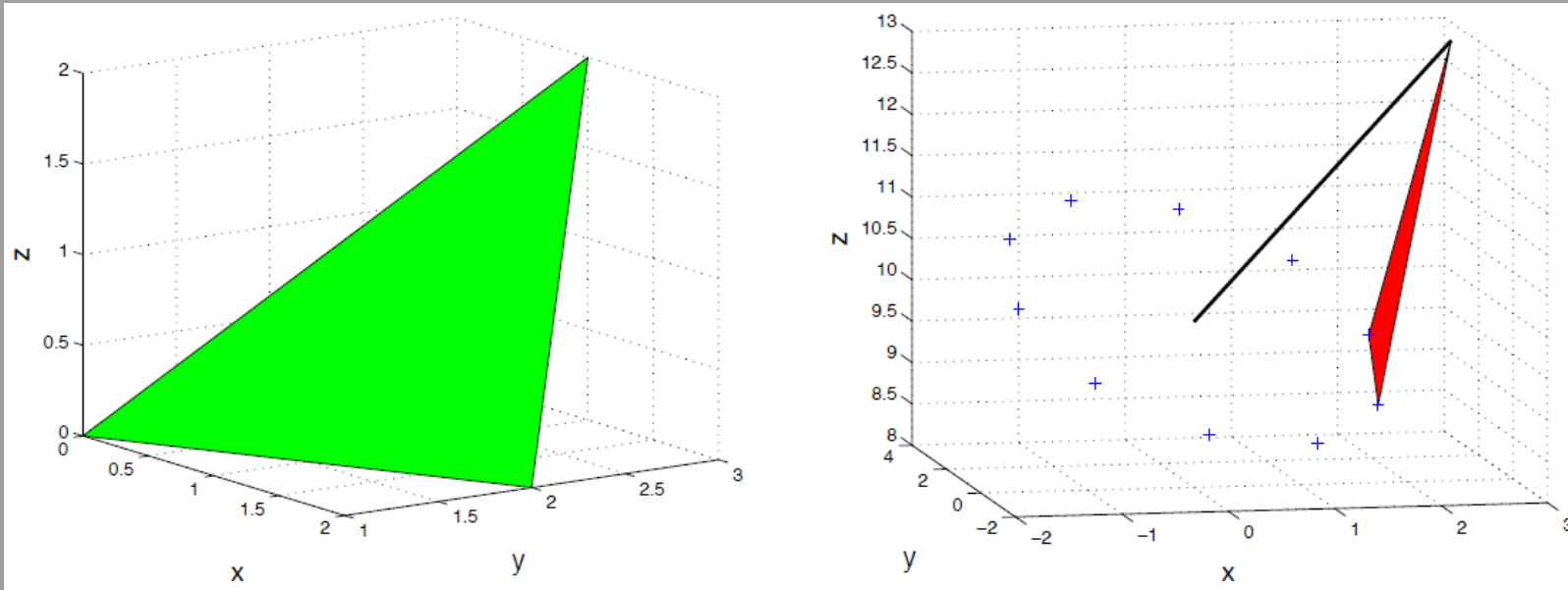


figura 2. Generación de triángulos: Izq. Ejemplo, Der. Triángulo del cono



Taller

d) General la superficie del cono, tal como se muestra en la figura.

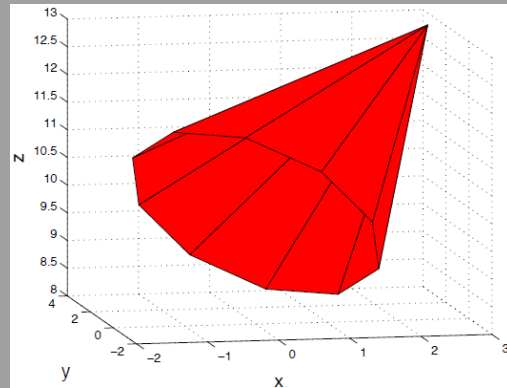


Figura 3. Superficie del cono

e) Dibujar la base del cono. Dibujar el cono aumentando el número de puntos a 100.

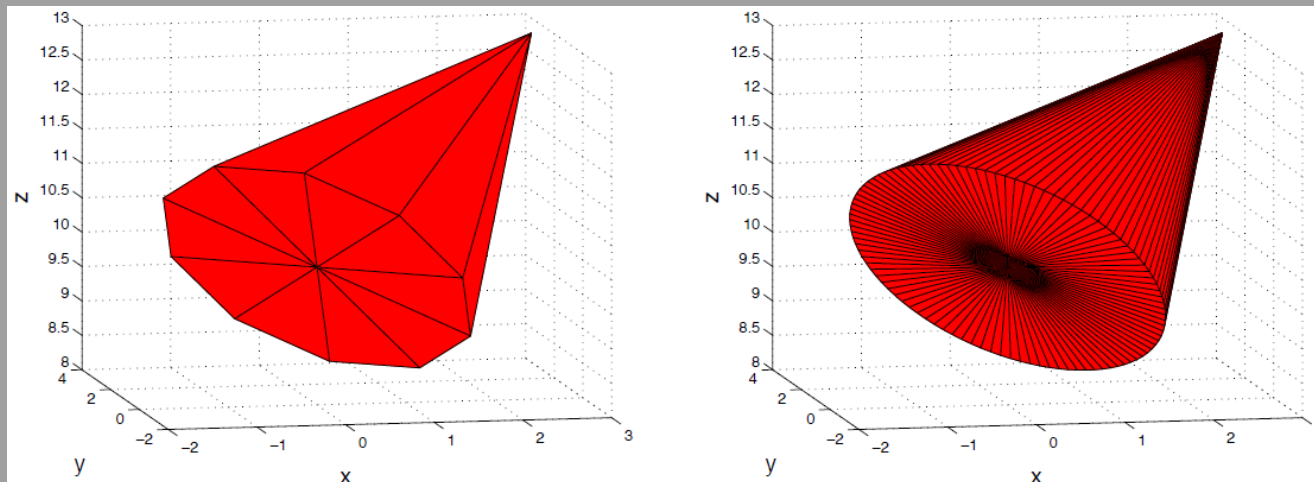


Figura 4. Cono generado con 10 puntos



Taller

3. De forma análoga al ejercicio 2, crear una función que dibuje una flecha teniendo en cuenta los siguientes parámetros de entrada:

- Vector de posición del origen de la flecha
- Vector que indica la dirección de la flecha (orientación).
- Longitud total de la flecha.
- Longitud de la punta de la flecha (diámetro de la base del cono)
- Diámetro del cilindro
- Diámetro de la base del cono
- Vector que indica el color ([R,G,B])



Taller

Ejemplo de dibujo de una flecha.

Código 3.

```
clear;
clc;
f7=figure(7);
clf
hold on;
set(f7,'Color',[1, 1, 1]);
grid on;
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
zlabel('z','FontSize',15);
view(10,28)
axis equal;
light;
pos = [1,1,1];
vec = [1,1,1];
tflecha = 2;
tpunta = 0.8;
dcil = 1;
dcono = 2;
color = [0,0,1];
dibujar_flecha3D(pos, vec,tflecha, tpunta,dcil,dcono, color);
```

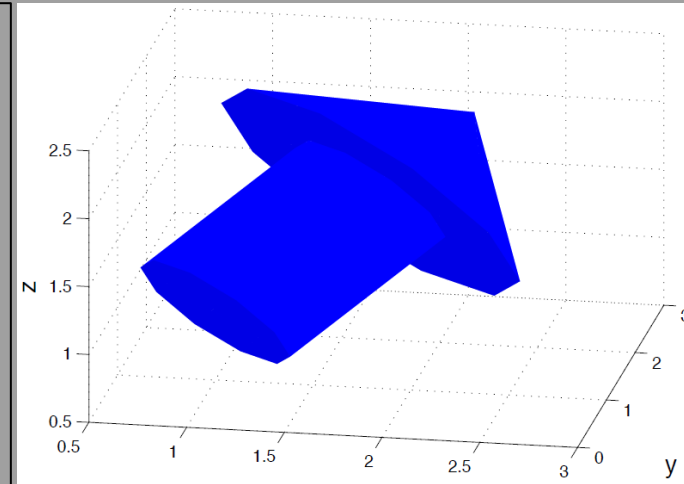


Figura 5. Flecha 3D.





Taller



4) Crear una función que dibuje un sistema coordenado. Teniendo en cuenta los siguientes parámetros de entrada:

dibujar_sistema_coordenadas_3D(A, L, G, texto)

- A: Matriz homogénea
- L: Longitud de los ejes coordenados
- G: Grosor de los ejes coordenados
- texto: nombre del sistema coordenado



Taller

Ejemplo del dibujo de un sistema de coordenadas

Código 4.

```
clear;
clc;
f8=figure(8);
clf;
hold on
set(f8,'Color',[1, 1, 1]);
grid on;
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
zlabel('z','FontSize',15);
view(28,54)
axis equal;
light;
Sn = [sqrt(2)/2 , -sqrt(2)/2 , 0 , 0 ; ...
      0          , 0          , 1 , 0 ; ...
      sqrt(2)/2 ,  sqrt(2)/2 , 0 , 0 ; ...
      0          , 0          , 0 , 1 ];
dibujar_sistema_coordenadas_3D(Sn, 10, 2, 'n');
```

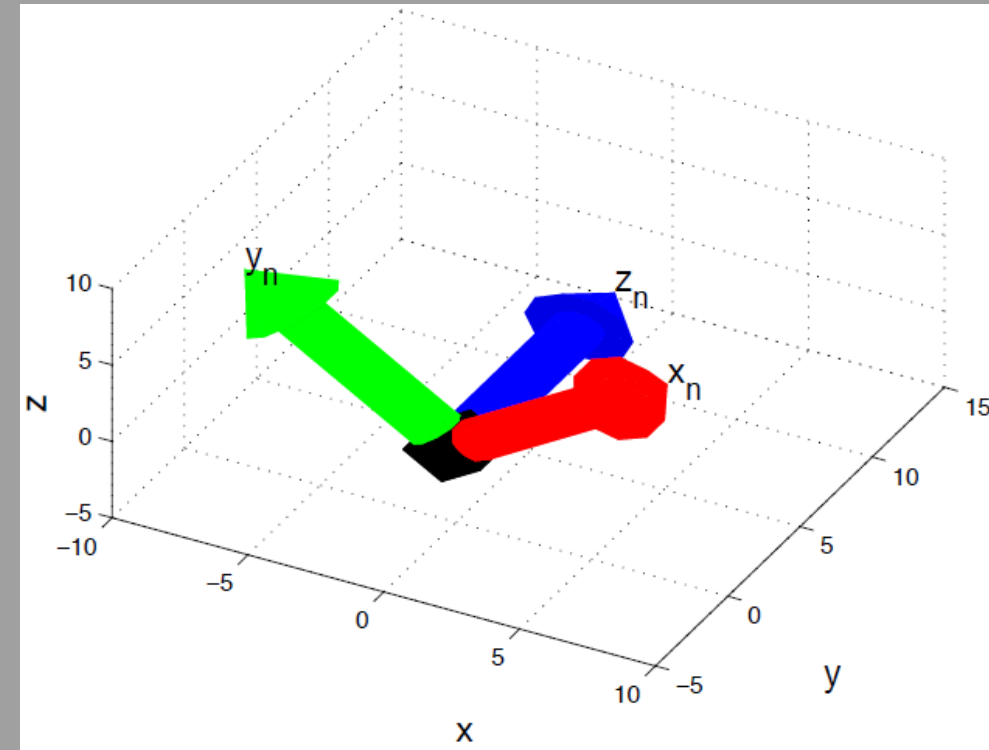


Figura 6. Sistema coordenado





Formando **líderes** para la
construcción de un nuevo
país en paz