UNTELS

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

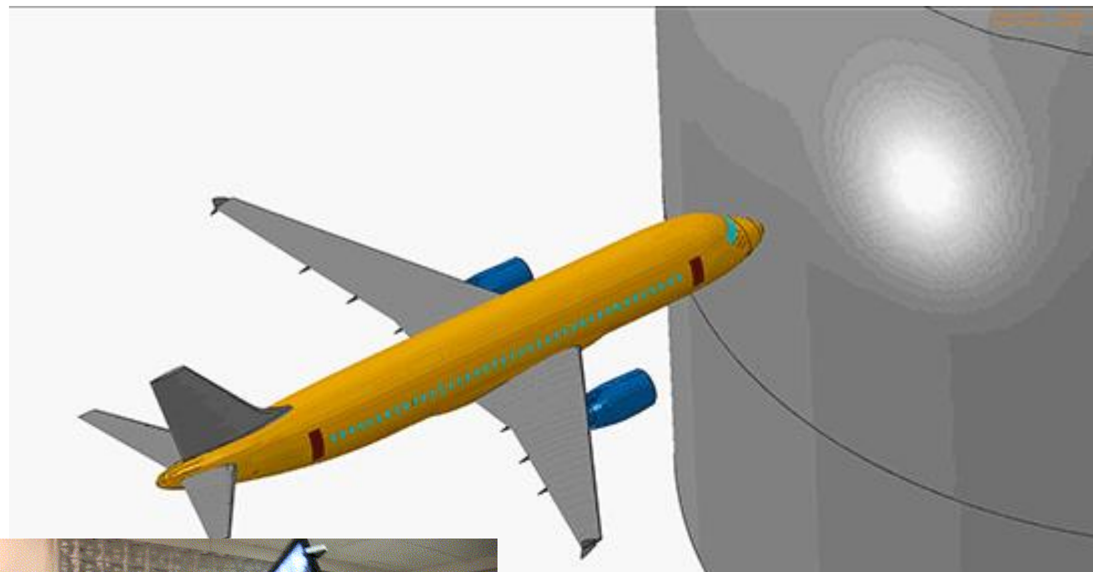
ALGORITMOS DE COMPUTACION GRAFICA

Lunes 13:00 - 16:20

Clase 03 Transformaciones Geométricas Bi-Dimensionales

Objetivo: Generar transformaciones geométricas bidimensionales.

Implementación de algoritmos de transformación.
Implementación de las coordenadas homogéneas.



MA. Juan Carlos Reátegui Morales
jreategui@untels.edu.pe
MBA-ISO 27001-ISO 9001-ISO 22301

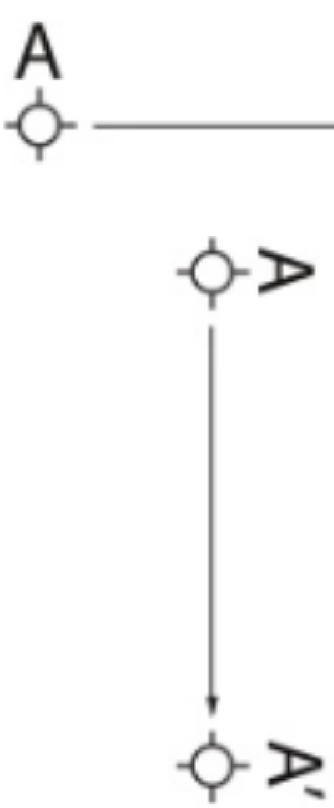
Todo lo que una persona puede imaginar, otras podrán hacerlo realidad.
Julio Verne

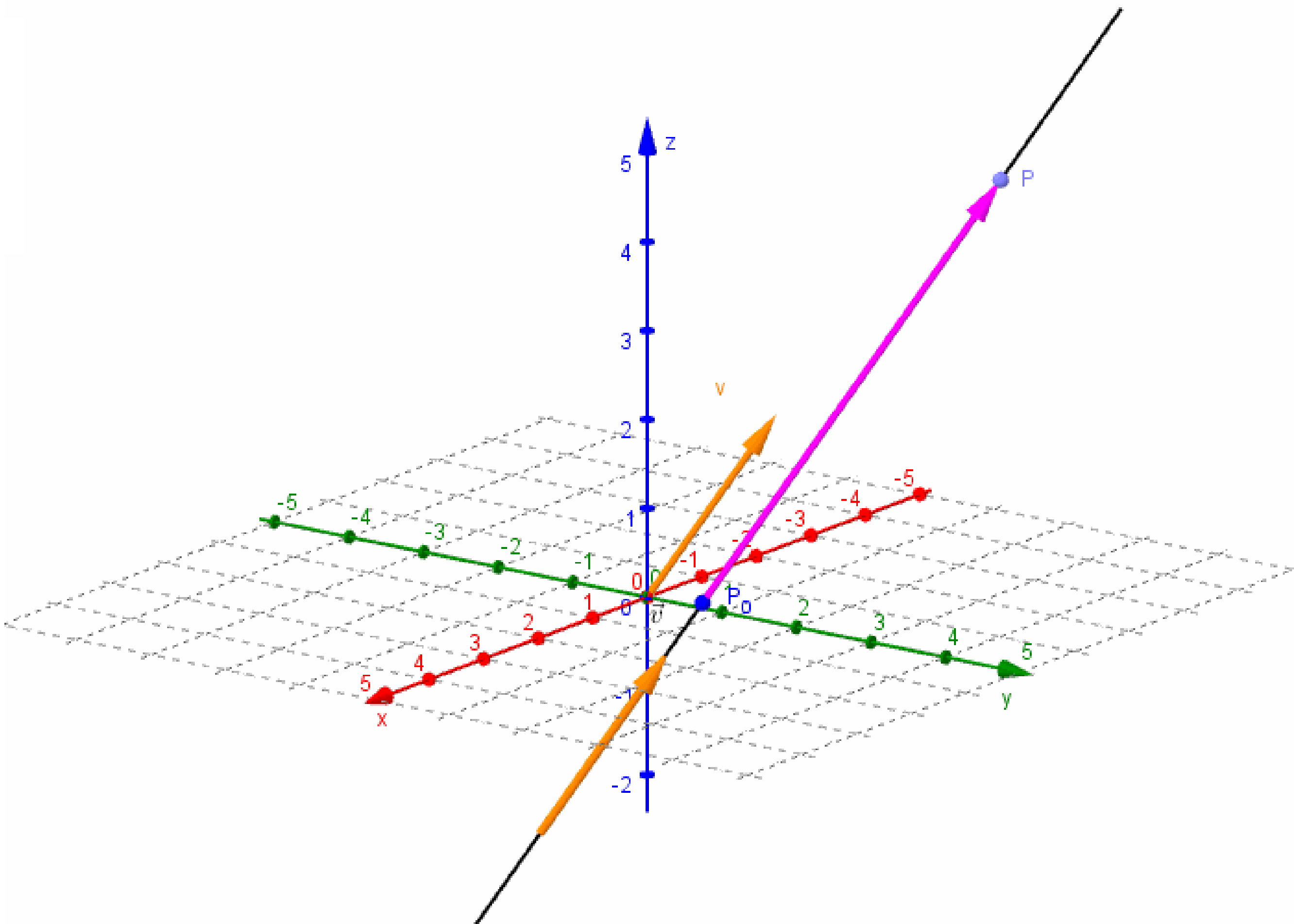
Transformaciones Bidimensionales

Transformación geométrica

Es una aplicación del plano tal que a cada punto de un plano le hace corresponder otro punto del mismo plano.

Son operaciones que permiten crear una nueva figura a partir de la posición original.





Transformaciones Bidimensionales

Una transformación de semejanza aplicada sobre una figura, es aquella que **no varía su verdadera forma después de la transformación**.

Para aplicar una transformación bidimensional de semejanza es necesario conocer como mínimo las coordenadas de dos puntos en ambos sistemas. Se mejora la precisión en la transformación, si los puntos se eligen lo más alejados posibles.

Una transformación de coordenadas bidimensional de semejanza consiste en tres pasos básicos:

- Giro.
- Cambio de escala.
- Traslación.

Supongamos dos sistemas de coordenadas ortogonales (X, Y) (x, y) independientes, donde un punto P está referido a los dos sistemas. Se van a deducir las relaciones entre las coordenadas de este punto en ambos sistemas, en los casos mencionados anteriormente.

Transformaciones Bidimensionales

Traslación de un punto **https://youtu.be/PA9DH1PJ9d8**

Es cuando se desplaza de en el plano en una nueva posición.

La traslación 2D

Sea P(*x*,*y*) se le traslada *Tx* unidades en el eje X, Ty en el eje Y

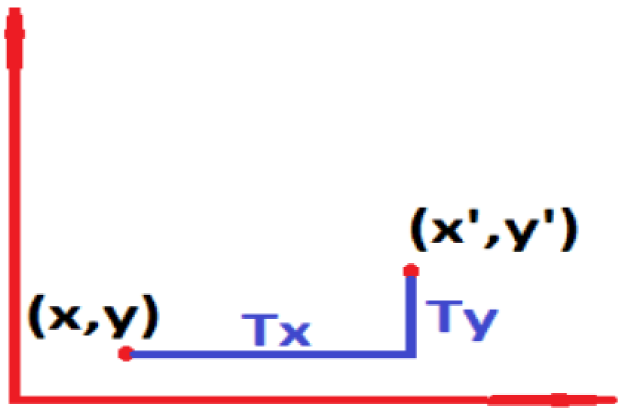
Por tanto las coordenadas del nuevo punto sería P'(*x'*,*y'*)

x' = x + Tx

y' = y +Ty

Por tanto la matriz de traslación quedaría de la siguiente forma:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Tx & Ty & 1 \end{bmatrix}$$



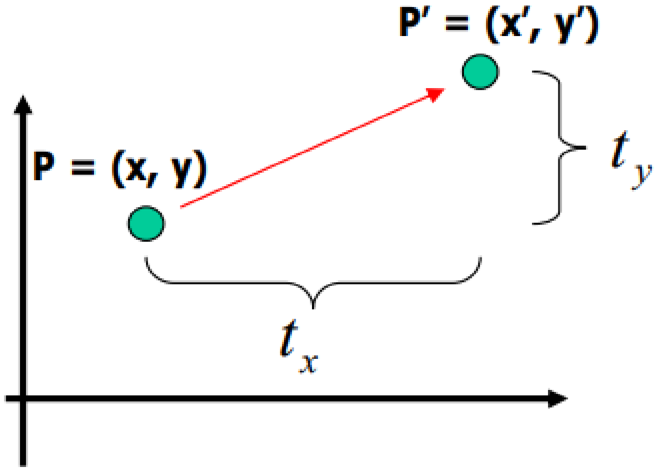
Transformaciones Bidimensionales

Traslación de un punto

https://youtu.be/z61YY5Ymt4I

- Reposiciona un objeto desplazándolo a las nuevas coordenadas

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

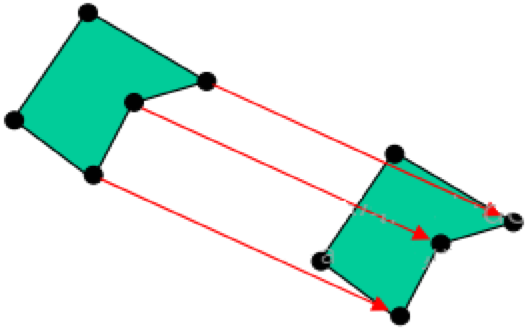


- En forma matricial:

$$P = (x, y) \qquad P' = (x', y') \qquad T = (t_x, t_y)$$

$$\boxed{P' = P + T}$$

- Es una transformación rígida → el objeto no se deforma
- Para trasladar líneas rectas trasladamos sólo sus extremos
- Para trasladar polígonos, trasladamos sólo sus vértices y redibujamos



Transformaciones Bidimensionales

Rotación de un punto

Es cuando se rota *respecto al origen* del plano, dando una nueva posición.

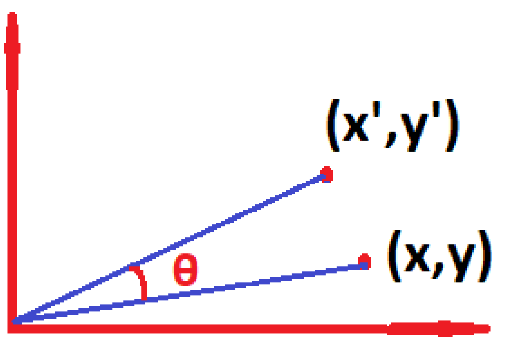
La Rotación 2D

Sea P(*x*,*y*) se le rota con un ángulo θ, por lo tanto las coordenadas del nuevo punto sería P'(*x'*,*y'*)

x' = x* coseno(θ) – y*seno(θ)
y' = x*seno(θ) + y* coseno(θ)

Por tanto la matriz de rotación quedaría de la siguiente forma:

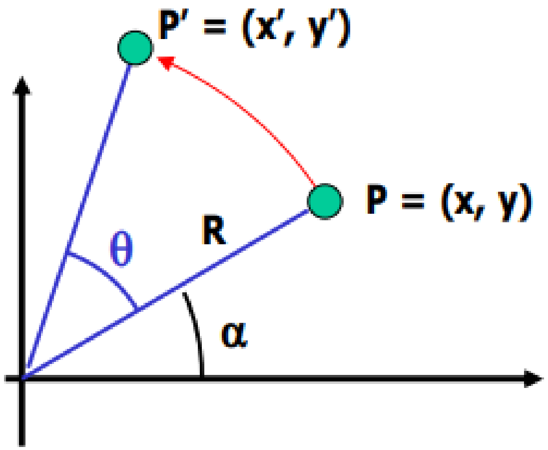
$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} \cos\theta & \operatorname{sen}\theta & 0 \\ -\operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Transformaciones Bidimensionales

Rotación de un punto

- La posición de un punto es rotada alrededor del origen de coordenadas
- ¿Cómo sacamos la fórmula para obtener P' a partir de P y del ángulo?



- Solución: expresándolo en polares

$$\begin{cases} x = R\cos\alpha \\ y = R\sin\alpha \end{cases} \qquad \begin{cases} x' = R\cos(\alpha+\theta) = \ldots = x\cos\theta - y\sin\theta \\ y' = R\sin(\alpha+\theta) = \ldots = x\sin\theta + y\cos\theta \end{cases}$$

- En forma matricial:

$$P = (x, y) \qquad P' = (x', y') \qquad R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

$$\boxed{P' = P \cdot R}$$

Transformaciones Bidimensionales

Rotación de un punto general

Es cuando se rota *respecto un punto especifico (Xc,Yc)* del plano, dando una nueva posición.

La Rotación 2D

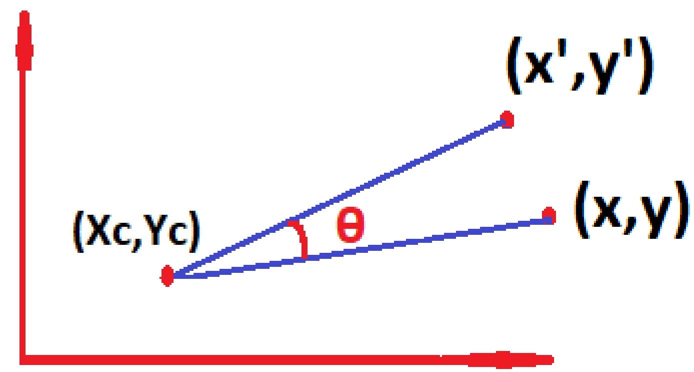
Sea P(*x*,*y*) se le rota con un ángulo θ desde una posición especifica (Xc,Yc), por lo tanto las coordenadas del nuevo punto sería P’(*x’*,*y’*)

x’ =xc+ (x-xc)* coseno(θ) – (y-yc)*seno(θ)

y’ = yc+ (y-yc)*seno(θ) + (y-yc)* coseno(θ)

Por tanto la matriz de rotación quedaría de la siguiente forma:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} (x-xc)\cos\theta & (y-yc)\sin\theta & 0 \\ -(y-yc)\sin\theta & (x-xc)\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Xc & Yc & 1 \end{bmatrix}$$



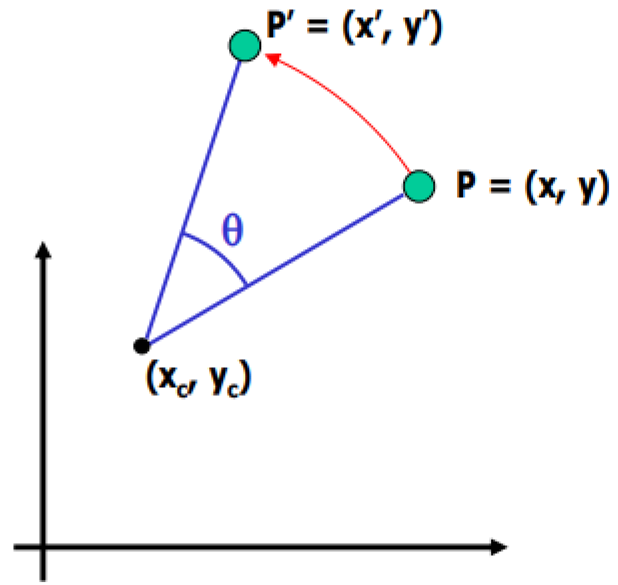
Transformaciones Bidimensionales

Rotación de un punto general

https://youtu.be/gY0cp-8FrI8

- ¿Cómo será la fórmula general cuando el punto sobre el que se rota no es el origen, sino un punto cualquiera (x_c, y_c)?

$$\begin{cases} x' = x_c + (x - x_c)\cos\theta - (y - y_c)\sin\theta \\ y' = y_c + (x - x_c)\sin\theta + (y - y_c)\cos\theta \end{cases}$$



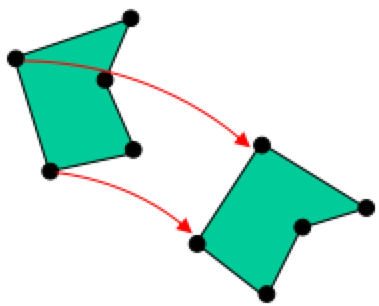
- Encontrar la forma matricial para este caso es un poco complicado

- Más tarde lo haremos de otra forma mucho más fácil

- Es una transformación rígida → el objeto no se deforma

- Para rotar líneas rectas rotamos sólo sus extremos

- Para rotar polígonos, rotamos sólo sus vértices y redibujamos



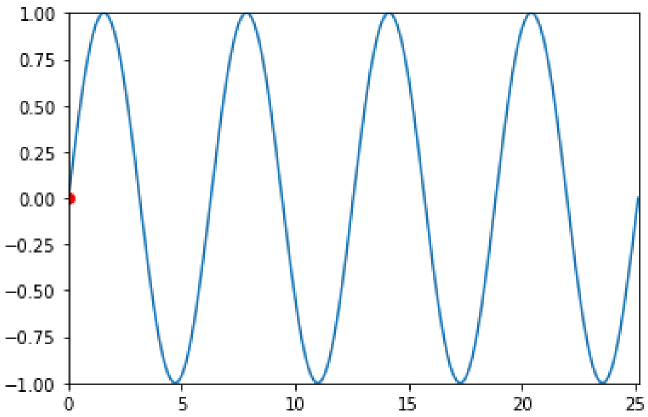
DESARROLLO DE APLICACIONES ALGORITMOS DE COMPUTACION GRAFICA

PRACTICA DE ALGORITMOS DE COMPUTACION GRAFICA

MA. Juan Carlos Reátegui Morales
jreategui@untels.edu.pe
MBA-ISO 27001-ISO 9001-ISO 22301

Transformaciones Bidimensionales

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
TWOPI = 2*np.pi                # Cambiar por 2-4-8-16
fig, ax = plt.subplots()
t = np.arange(0.0, TWOPI, 0.001)
s = np.sin(t)
l = plt.plot(t, s)
ax = plt.axis([0,TWOPI,-1,1])
redDot, = plt.plot([0], [np.sin(0)], 'ro')
def animate(i):
 redDot.set_data(i, np.sin(i))
 return redDot,
# create animation using the animate() function
myAnimation = animation.FuncAnimation(fig, anima
te, frames=np.arange(0.0, TWOPI, 0.1), \
 interval=10, blit=True, repeat=True)
plt.show()
```



Sumas y Restas de Imágenes

(316, 295, 3)

(316, 295, 3)

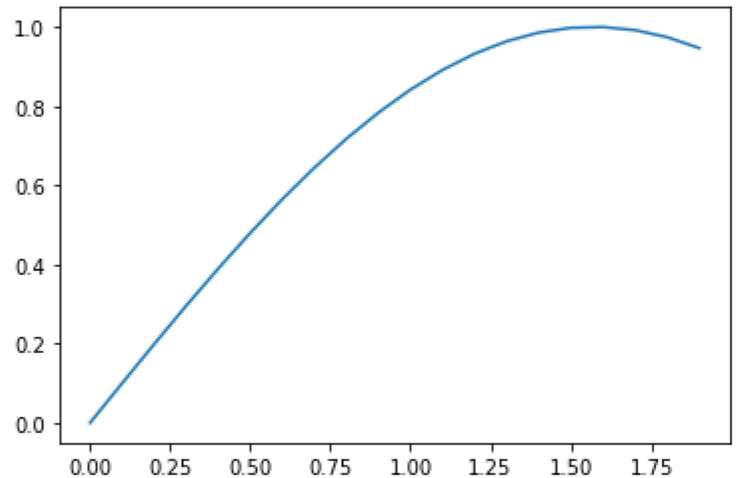
```
#------------------------------------------------------------------------
#Suma dos imagenes y resta dos imagenes
#Autor:
#Fecha: 02-05-2022
#------------------------------------------------------------------------
# import the cv2 library
import cv2
from google.colab.patches import cv2_imshow
# Lee imagenes
camarografo = cv2.imread('cameraman.tif')
calculadora = cv2.imread('keyb.tif')
# Imprime las imagees
print(camarografo.shape)
cv2_imshow(camarografo)
print(calculadora.shape)
cv2_imshow(calculadora)
# Como las filas y columnas difieren, compatibilizamos la imagen del camarografo
up_width = 295
up_height = 316
up_points = (up_width, up_height)
camarografo_x = cv2.resize(camarografo, up_points, interpolation= cv2.INTER_LINEAR)
#Imprimo la imagen con su nueva estructura
print(camarografo_x.shape)
cv2_imshow(camarografo_x)
# Sumamos las matrices
suma = cv2.add(camarografo_x,calculadora)
# Imprimimos la matriz suma
print(suma.shape)
cv2_imshow(suma)
# Restamos las matrices
resta = cv2.subtract(camarografo_x,calculadora)
# Imprimimos la matriz resta
print(resta.shape)
cv2_imshow(resta)
# waitKey() espera que presiones una tecla para cerrar la ventana y se da 0 sigue el loop
cv2.waitKey(0)
# cv2.destroyAllWindows() destruye la ventana creada
cv2.destroyAllWindows()
```

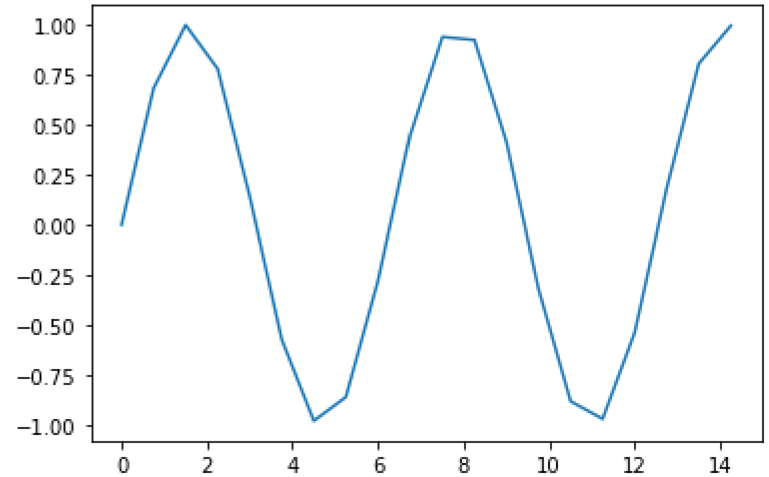
Ejercicios Asíncronos

```
import math
import numpy as np
from matplotlib import pyplot as plt
x= np.array(range(20))*0.1
print ("X=",x)
y= np.zeros(len(x))
print ("Y=",y)
print ("Función Seno")
for i in range(len(x)):
  y[i]=math.sin(x[i])
# Creamos el gráfico
plt.plot(x,y)
plt.show()
```

Lo hacemos variar de a 0.25,0.50,0.75

```
X= [0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
 1.8 1.9]
Y= [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Función Seno
```

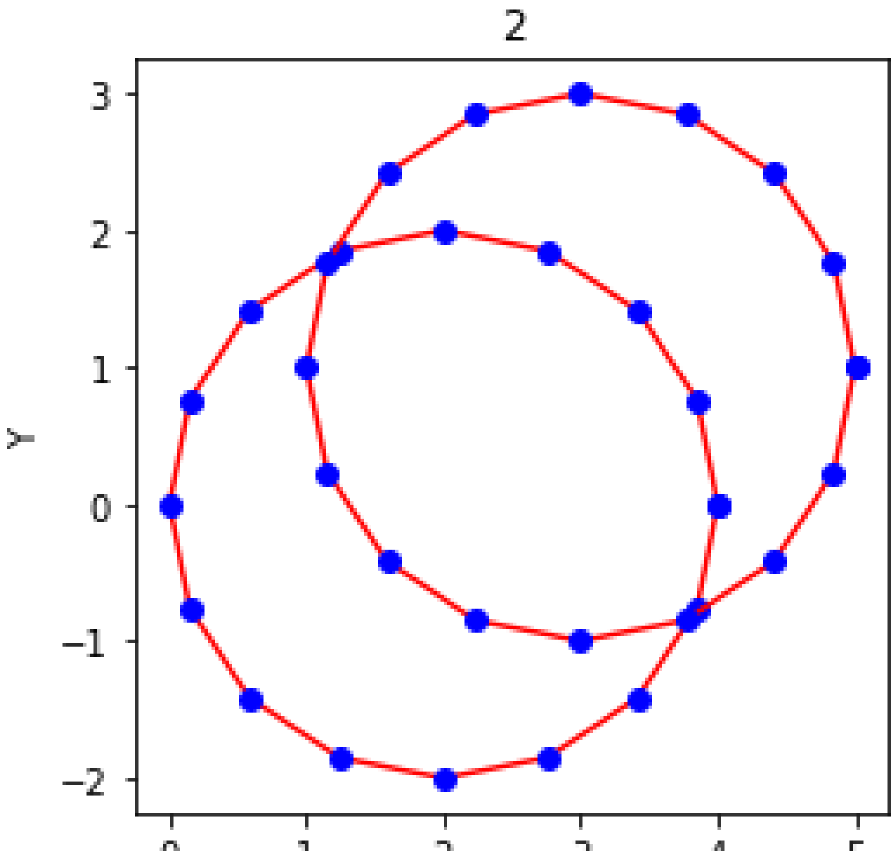




Ejercicios Asíncronos

```
#-----------------------------------
# Grafica circulos en diferentes posiciones
#Autor:
#Fecha: 02-05-2022
#-----------------------------------
import numpy as np
from matplotlib import pyplot as plt
num_segmentos = 16
rad = 2
cx = 2
cy = 0
n=0
# Cuando veces muestro el grafico
coleccion = [1,2,3,4]
for e in coleccion:
  angulo = np.linspace(0, 2*np.pi, num_segmentos+1)
  x = rad * np.cos(angulo) + cx
  y = rad * np.sin(angulo) + cy
  cx= cx + 1
  cy= cy + 1
  n= n + 1

  plt.plot(x, y, color="red", markersize=1)
  plt.title(n)
  plt.xlabel("X")
  plt.ylabel("Y")
  plt.gca().set_aspect('equal')
  plt.grid()
 # plt.show()
```



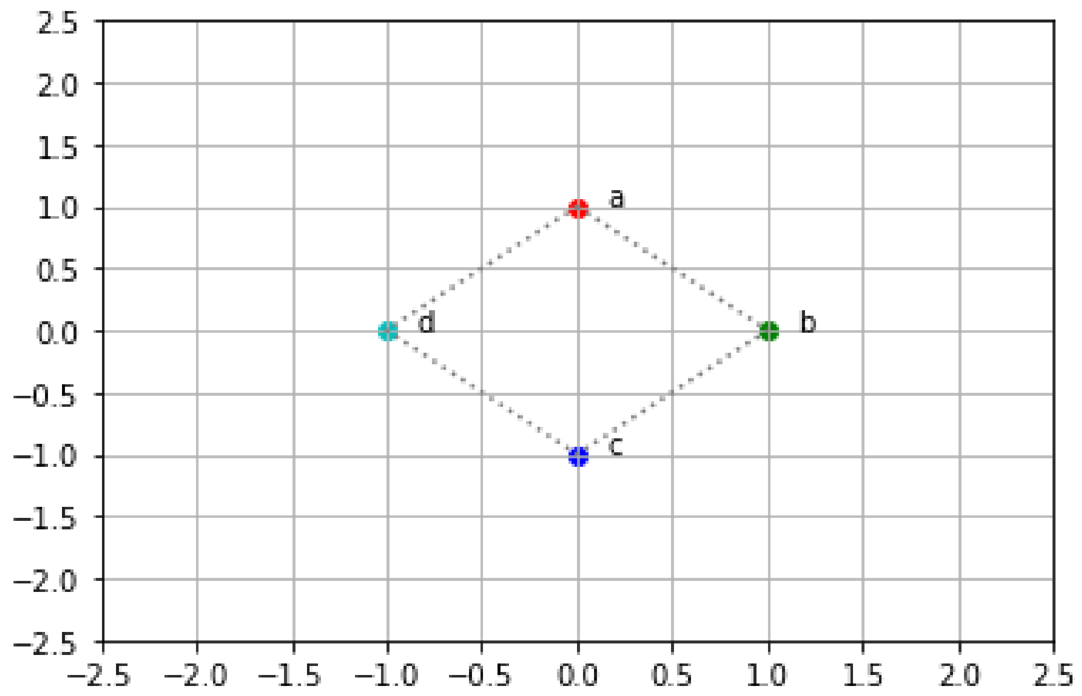
Ejercicios Asíncronos

```
# points a, b and, c
a, b, c, d = (0, 1, 0), (1, 0, 1), (0, -1, 2), (-1, 0, 3)

# matrix with row vectors of points
A = np.array([a, b, c, d])

# 3x3 Identity transformation matrix
I = np.eye(3)

color_lut = 'rgbc'
fig = plt.figure()
ax = plt.gca()
xs = []
ys = []
for row in A:
    output_row = I @ row
    x, y, i = output_row
    xs.append(x)
    ys.append(y)
    i = int(i) # convert float to int for indexing
    c = color_lut[i]
    plt.scatter(x, y, color=c)
    plt.text(x + 0.15, y, f"{string.ascii_letters[i]}")
xs.append(xs[0])
ys.append(ys[0])
plt.plot(xs, ys, color="gray", linestyle='dotted')
ax.set_xticks(np.arange(-2.5, 3, 0.5))
ax.set_yticks(np.arange(-2.5, 3, 0.5))
plt.grid()
plt.show()
```



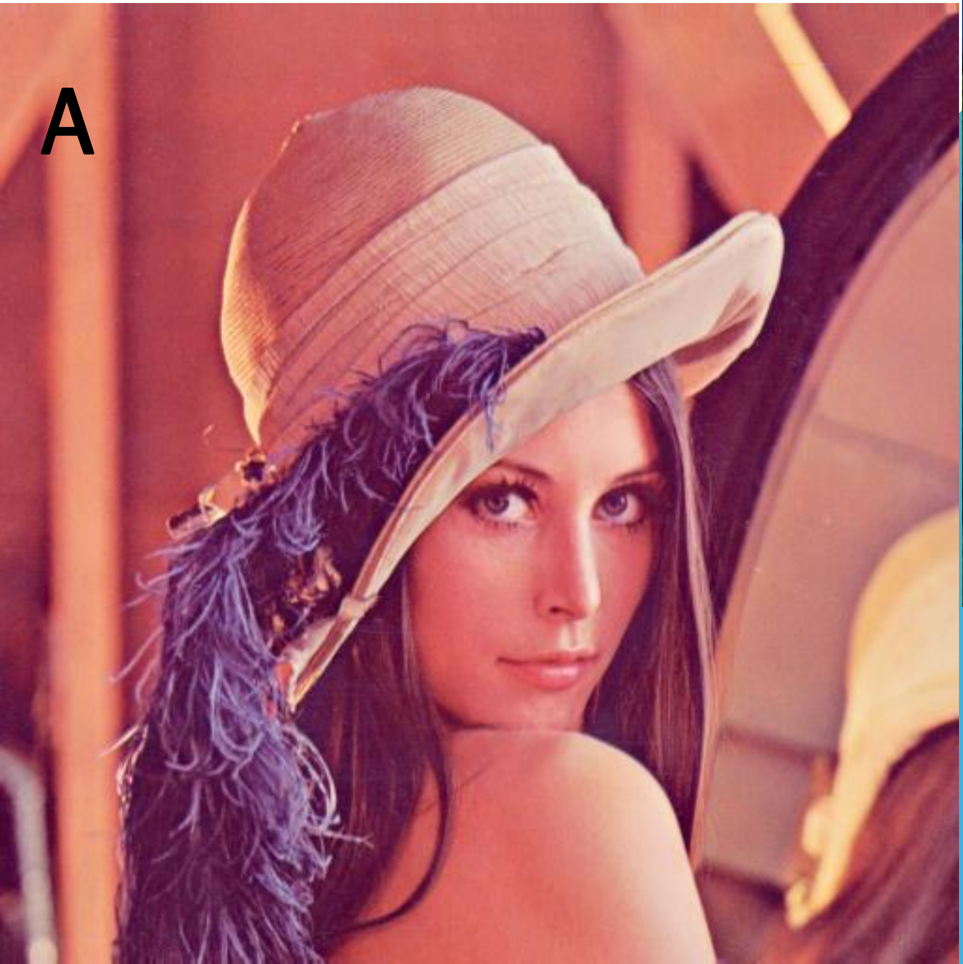
Ejecutar y explicar la figura

Ejercicios Asícronos (Recortes de imágenes)

Generar :

A + B

A - B y B - A



B

Control de Aprendizaje

ACG	Horario: 13.00 - 16.20		

Lunes

N°	Código	Alumno	Proyecto
1	1923050072	ALANYA VILLAR JOEL EDWIN	Videojuegos con Unity
2	1923110371	ALVA CHANTA EDSON ALCIDES	Reconoc. De Imágenes en Medicina
3	2014100223	ANCHAYHUA GUTIERREZ DAVID ANDRE	Animaciones con Anime
4	2014101343	AZAÑERO ESPINOZA WALDIR YSAI	Identifica personas con Python
5	1913010968	CARRASCO CHINCHAY HENRY ELI	Identifica personas con Python
6	1913010160	CCACCYA HUAMAN ANTONY	RV-RA en Turismo
7	1913010916	CHAVEZ GAMARRA JOSE CARLOS	RV-RA en Turismo
8	1813011317	DIAZ SEMINARIO DANIEL OMAR	Cuadro de Mando Integral (CMI)
9	1923110141	FLORES CHAMBA JOSE	Reconoc. De Imágenes en Medicina
10	1913010296	FLORES HERRERA JULIO CHRISTIAN	RV-RA en Turismo
11	1813011644	GOMEZ HUAMANI STEVE EDWARD	Identifica personas con Python
12	1913110530	HUANCAS LEUYACC ANSELMO JUNIOR	RV-RA en Turismo
13	1813011665	LEANDRO BLAS LUIGGI ANDERSON	Animaciones con Anime
14	1923010512	MOTTA MENDOZA MIGUEL ANGEL	Reconoc. De Imágenes en Medicina
15	1823110188	NOBLEJAS SAAVEDRA JORDAN MOISES	Animaciones con Anime
16	2016200172	PONCE SUSANIBAR ALONSO GAVINO	Cuadro de Mando Integral (CMI)
17	2017110656	QUISPE CUPE JORDY EUSEBIO	Videojuegos con Unity
18	1923010511	SAYAS DE LA VEGA PIERO GABRIEL	Identifica personas con Python
19	2008100166	TORRES BARRIENTOS CARLOS JOSSIMAR	Cuadro de Mando Integral (CMI)
20	1923110436	YAURICASA MENDOZA MIGUEL ANGEL	Reconoc. De Imágenes en Medicina
21	2016200249	ZEVALLOS TORRES DIEGO LEONEL	Videojuegos con Unity
22	2012100065	SIlvestre Abarca Jorge Javier	Animaciones con Anime

AVANCES DEL PROYECTO

- ✓ **Leemos los paper de publicaciones científicas y argumentar su importancia.**

- ✓ **Buscamos en la web, cuando menos 10 documentos de proyectos referentes al tema (antigüedad máxima de 5 años).**

- ✓ **Lo ponemos en el drive compartido de la UNTELS.**

…

TRANSFORMACIÓN DIGITAL

Para la práctica Calificada 01 (Clase 04)

Una Landing Page (página de aterrizaje) es una página dentro de un sitio web, desarrollada con el único objetivo de convertir los **visitantes en Leads o prospectos de ventas** por medio de una oferta determinada. Generalmente tiene un diseño más sencillo con pocos enlaces e informaciones básicas sobre la oferta, además de un formulario para realizar la conversión.

Con efectos especiales:

- Hojas de estilo en cascada (del ingles Cascading Stylesheets CSS).
- BootStrap.

https://www.youtube.com/watch?v=NmaSiRGeROs

Control de Aprendizaje

¿Qué es una transformación bidimensional?. Para que sirven.

¿Qué importancia tienen las transformaciones bidimensionales?

¿Cómo podemos generar valor con aplicaciones de transformación bidimensional?

¿Qué opina sobre las publicaciones científicas (paper)?

¿Qué puntos seria importante investigar en el contexto de Computación Gráfica?

¿Qué importancia tienen la Landing Page para un Ingeniero de Sistemas?

¿Qué importancia tienen las sentencias CSS (Hojas de Estilo en Cascada) y el Framework Bootstrap para un Ingeniero de Sistemas?. ¿Qué utilidad se le puede dar?

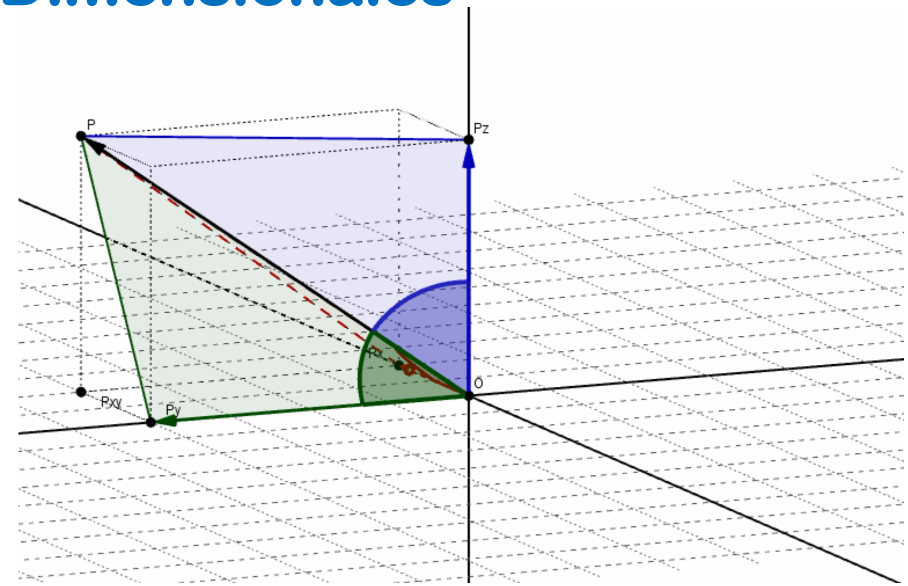
ALGORITMOS DE COMPUTACION GRAFICA

Lunes 13:00 - 16:20

Clase 04 Transformaciones Geométricas Tri-Dimensionales

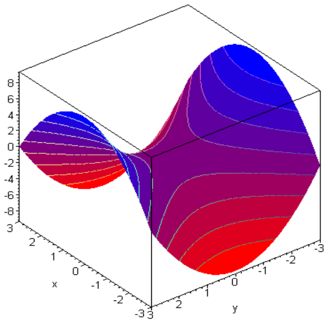
Objetivo: Generar transformaciones geométricas tridimensionales.

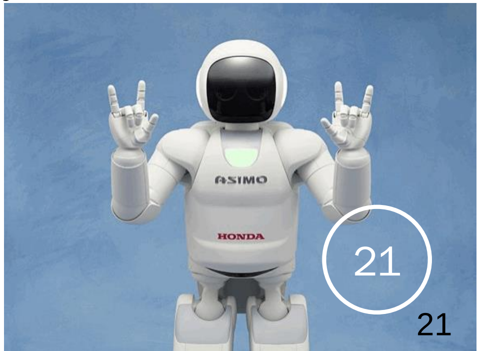
Implementación de algoritmos de transformación.
Implementación de las coordenadas homogéneas.



La mejor forma de predecir el futuro, construirlo.
Alan Kay

"¡ Que extraño: Cuando mas me esfuerzo, más suerte tengo!"
Henry Ford





MA. Juan Carlos Reátegui Morales
jreategui@untels.edu.pe
MBA-ISO 27001-ISO 9001-ISO 22301

Muchas gracias…