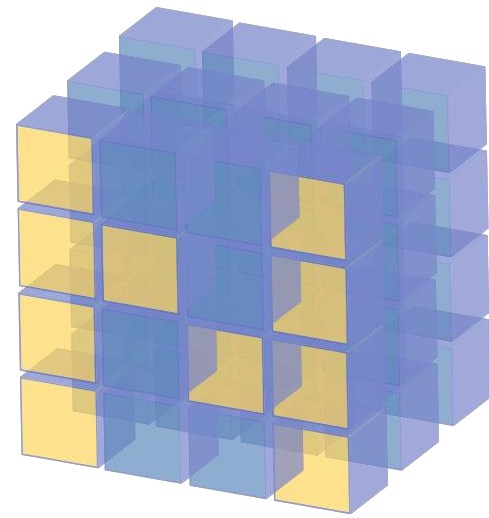


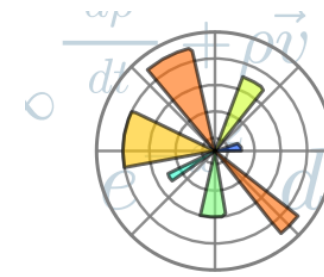
# PROCESAMIENTO DIGITAL DE IMAGENES



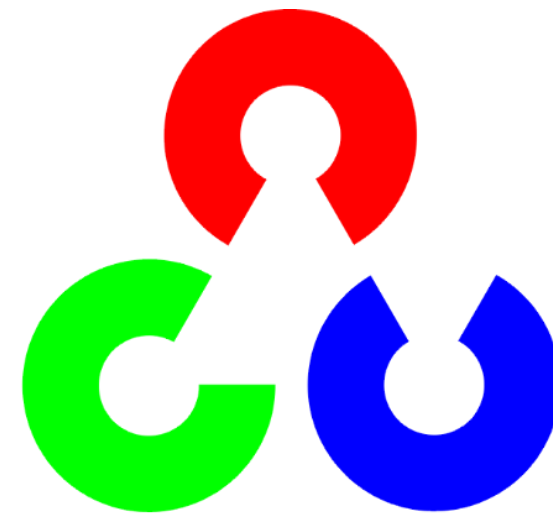
# Módulos principales



NumPy



matplotlib



OpenCV

`pip install opencv-python`



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN  
DE DESARROLLO TECNOLÓGICO

# Introducción a OpenCV

Dar una introducción al mundo del procesamiento de imágenes con OpenCV

Definir las librerías que usaremos en este curso

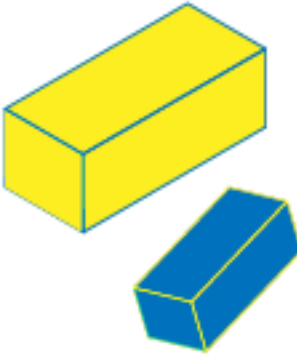
## ¿Que es OpenCV?

OpenCV (Open Source Computer Vision) es una librería de programación de código abierto dirigida principalmente a la visión por computador en tiempo real, desarrollada por la división rusa de Intel en el centro de Nizhni Nóvgorod.

OpenCV permite desarrollar en C, C ++ o Python y es compatible con el IDE QT Creator y sus correspondientes librerías QT.

## Las áreas de aplicación de OpenCV incluyen:

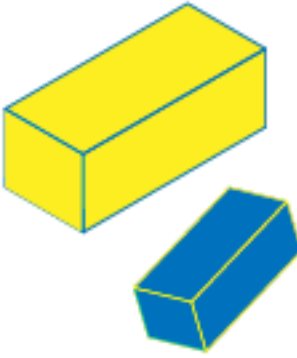
- Odometría visual
- Sistema de reconocimiento facial
- Reconocimiento de gestos
- La interacción hombre-ordenador (HCI)
- Robótica móvil
- Comprensión del movimiento
- Identificación del objeto
- Segmentación y reconocimiento
- Visión estereoscópica: percepción de profundidad desde 2 cámaras
- Estructura de movimiento (SFM)
- Rastreo de movimiento
- Realidad aumentada
- Aprendizaje del árbol de decisiones
- Algoritmo de maximización de expectativas
- Algoritmo de vecino más cercano k
- Clasificador Bayes
- Redes neuronales artificiales
- Soporte de máquinas vectoriales
- Redes neuronales profundas



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

## TEMAS QUE VEREMOS HOY

- Imagen
- Acceso a pixel
- Leer una imagen
- Mostrar una imagen
- Guardar una imagen
- Uso de matplotlib
- El objeto imagen
- Slicing en imágenes
- Videos con OpenCV
- Reproducción de video
- Guardado de un video



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES



# IMÁGEN

Es una representación del mundo físico que tiene información importante, la cual es captada mediante un proceso de muestreo, generalmente por medios electrónicos



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# IMÁGEN DIGITAL

La imagen digital es el archivo resultante de una discretización de una imagen natural ó sintética, en elementos de imagen llamada pixeles.

González A.

Una imagen digital puede definirse como una función bidimensional  $f(x,y)$  donde  $x$  e  $y$  son coordenadas en el plano y la amplitud  $f$  es llamada intensidad o nivel de gris en ese punto. Garcia S. (2008)

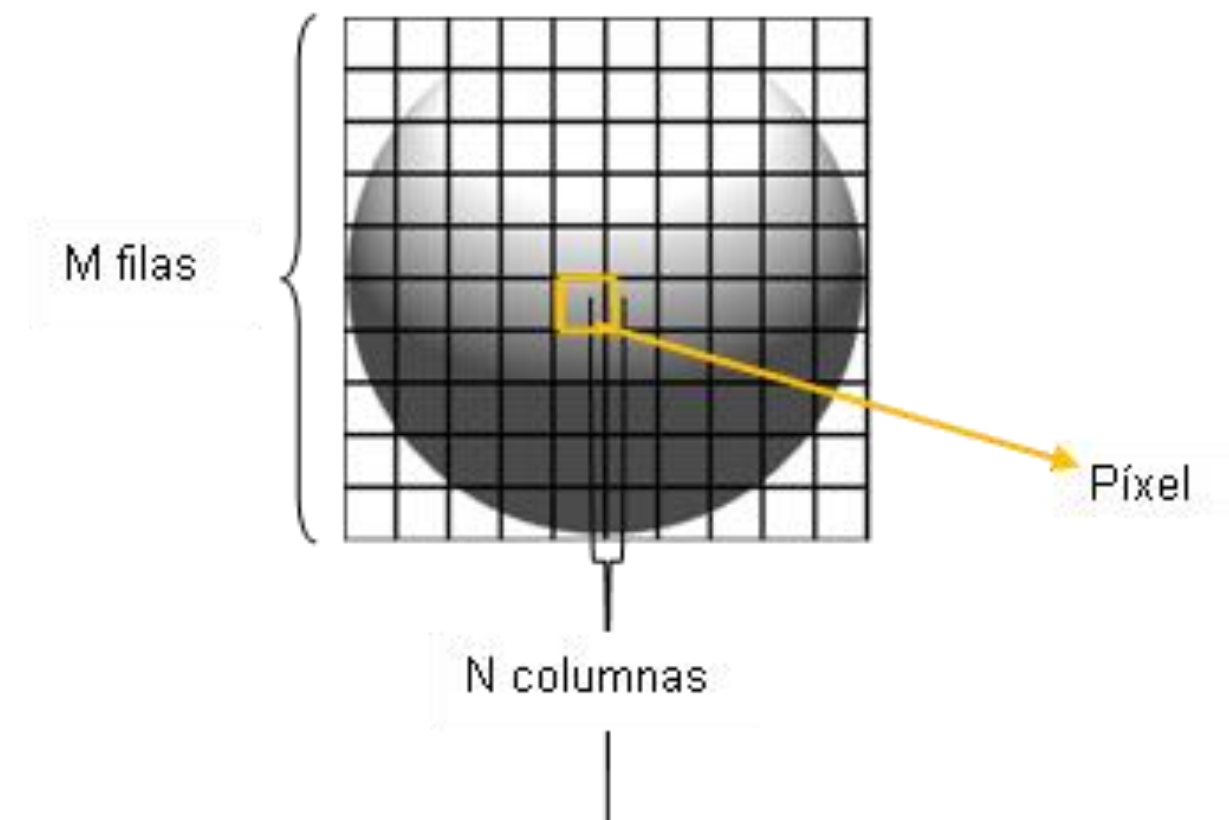


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES



# PIXEL

- El pixel es la unidad más pequeña y diminuta de una imagen digital.
- Es un espacio en la memoria de la computadora donde se almacena un numero que representa la definición del color y el brillo de una parte de la imagen
- Cada pixel define solamente un color
- El numero de pixel define la información que contiene una imagen

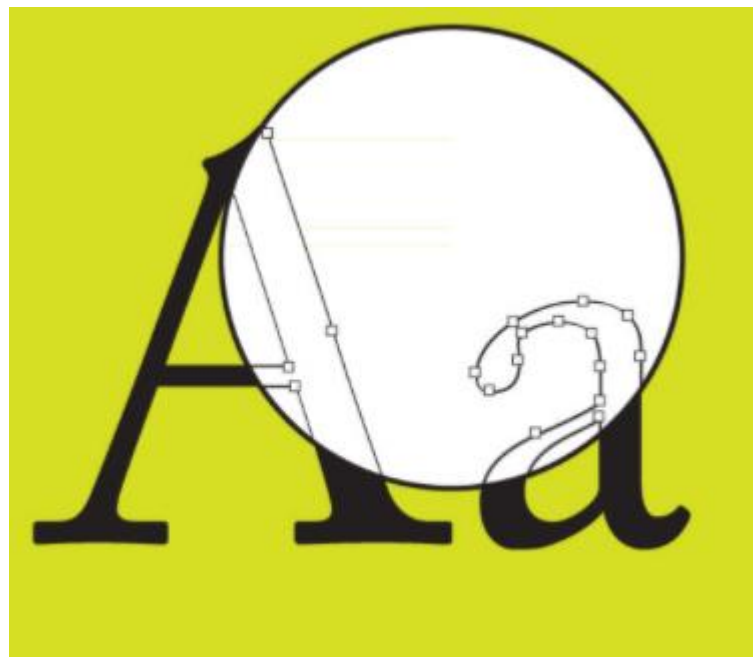


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# CLASIFICACIÓN DE LAS IMAGENES

## IMAGENES VECTORIALES

- Las imágenes vectoriales están compuestas por entidades geométricas simples : segmentos y polígonos básicamente
- Al estar compuesta por entidades geométricas simples, las imágenes vectoriales se pueden cambiar de escala, a para ampliarlas o reducirlas ,sin que la imagen pierda calidad.

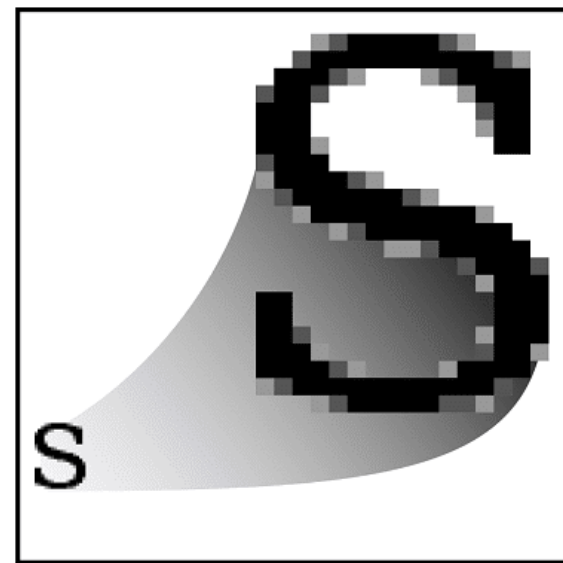


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# CLASIFICACIÓN DE LAS IMAGENES

## IMAGENES VECTORIALES

- Esta es su gran ventaja, porque proporcionan siempre imágenes de colores planos con contornos limpios, sin importar el tamaño al que se muestran.
- Al ampliar la imagen no se pierde la calidad en los bordes, se puede observar más detalle de esta
- Entre los programas utilizados para realizar esta clase de imágenes se destacan: Corel Draw, Illustrator, Autocad , Flash



.jpeg .gif .png



.svg

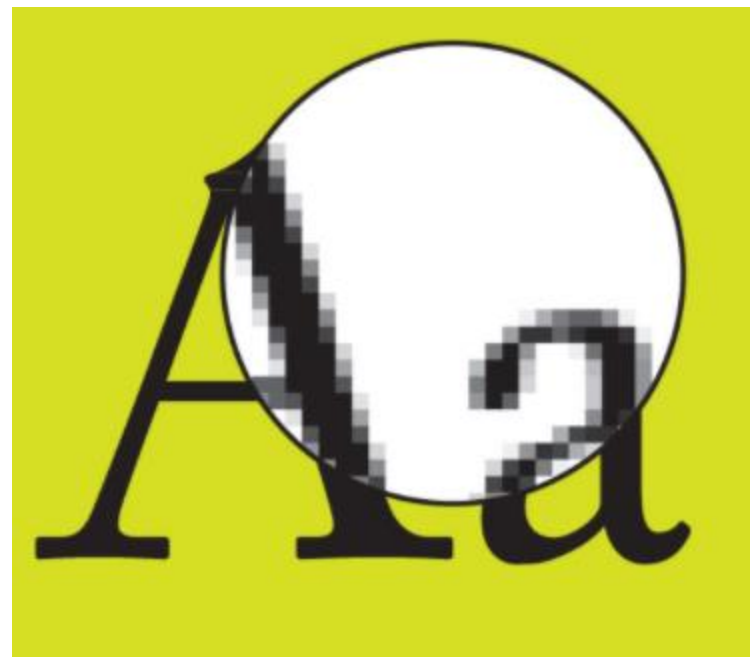


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# CLASIFICACIÓN DE LAS IMAGENES

## IMAGENES BITMAP

- Las imágenes de mapa de bits están construidas mediante una gran cantidad de cuadraditos, llamados pixel
- Alternativa ideal para reproducir objetos sutilmente iluminados y escenas con gran variación tonal
- Las imágenes bitmap no permiten el cambio de escala



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES



# CLASIFICACIÓN DE LAS IMAGENES

## IMAGENES BITMAP

- Las imágenes de mapa de bits están construidas mediante una gran cantidad de cuadraditos, llamados pixel
- Alternativa ideal para reproducir objetos sutilmente iluminados y escenas con gran variación tonal
- Las imágenes bitmap no permiten el cambio de escala

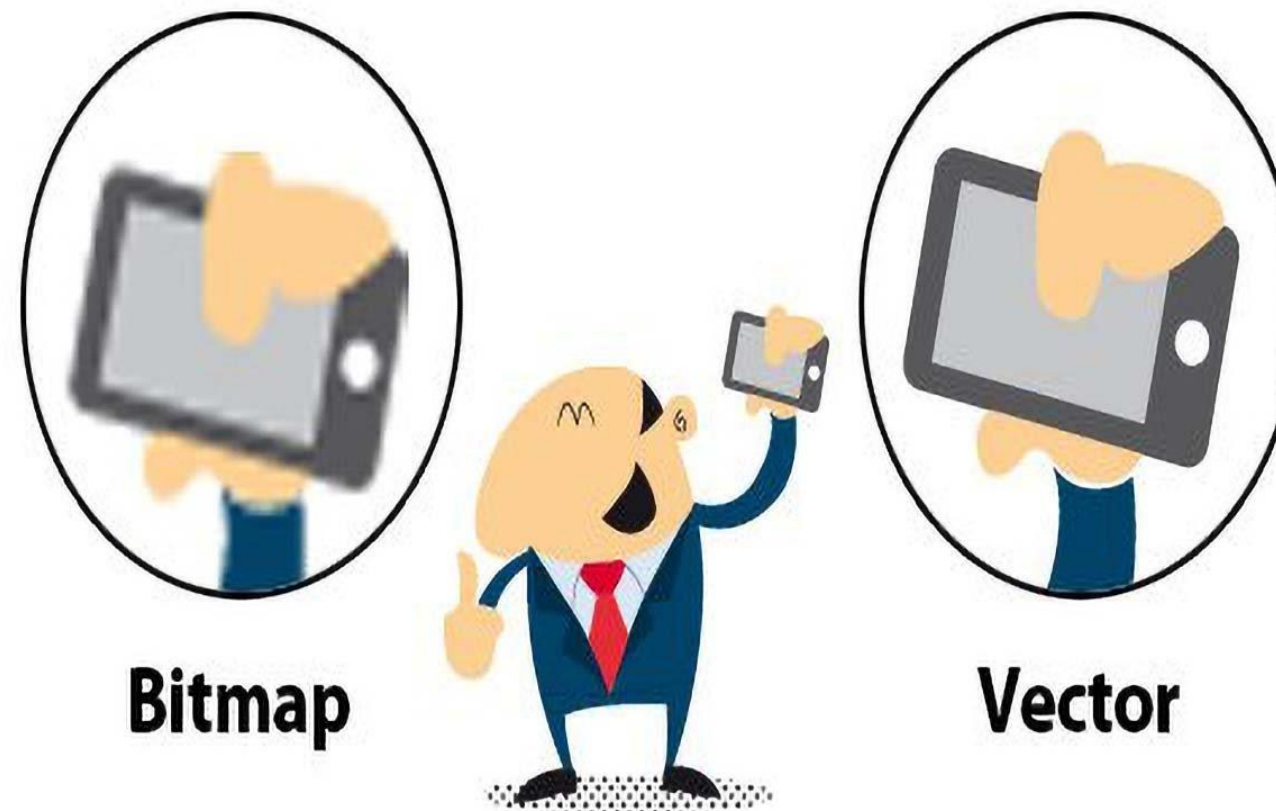


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# CLASIFICACIÓN DE LAS IMAGENES

## IMAGENES BITMAP

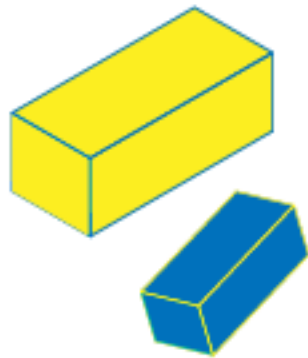
- Este efecto que se conoce con el nombre de pixeleado se hace mas evidente en las líneas curvas y en las zonas en las que hay cambios bruscos de luminosidad
- Los programas mas utilizados para generar o editar este tipo de imágenes bitmap son el famoso Photoshop de Adobe, Paint Brush de Windows y con el Photopaint de Corel



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES



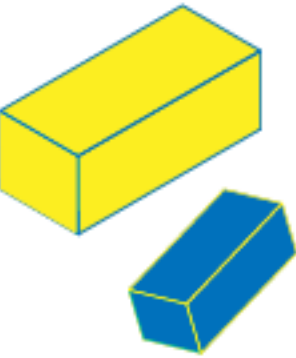
# FORMATOS DE IMÁGENES RASTER O BITMAP



FORMATO	SIGNIFICADO	EXTENSION	DEFINICION
Bit Map	Mapa de Bits		Es el formato tradicional para los usuarios de Windows. Es usado con propósitos generales como : edición de imágenes y tapiz del escritorio de Windows



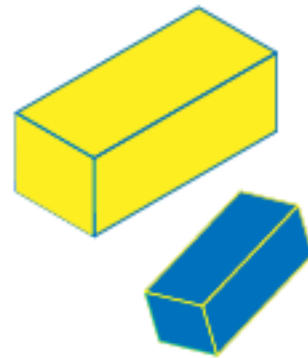
# FORMATOS DE IMÁGENES RASTER O BITMAP



FORMATO	SIGNIFICADO	EXTENSION	DEFINICION
Graphic Interchange Format	Formato de intercambio de graficos		<p>Se trata de un formato de archivo muy compacto de uso muy común en las paginas web. Su principal desventaja es que limita las imágenes a solo 256 colores lo cual puede afectar la calidad de la imagen en pantalla. Este formato incluye 3 características:</p> <ul style="list-style-type: none"><li>• Transparencia</li><li>• Entrelazado</li><li>• Animación</li></ul>



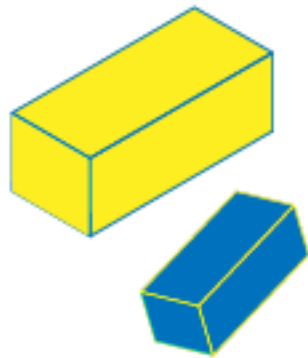
# FORMATOS DE IMÁGENES RASTER O BITMAP




FORMATO	SIGNIFICADO	EXTENSION	DEFINICION
Joint Photographic Experts Group	Grupo Unido de Expertos en Fotografia	 	Creado por el grupo Unido de expertos en Fotografía. Es un formato de archivo comprimible con posibilidades de escalamiento para producir archivos reducidos.



# FORMATOS DE IMÁGENES RASTER O BITMAP



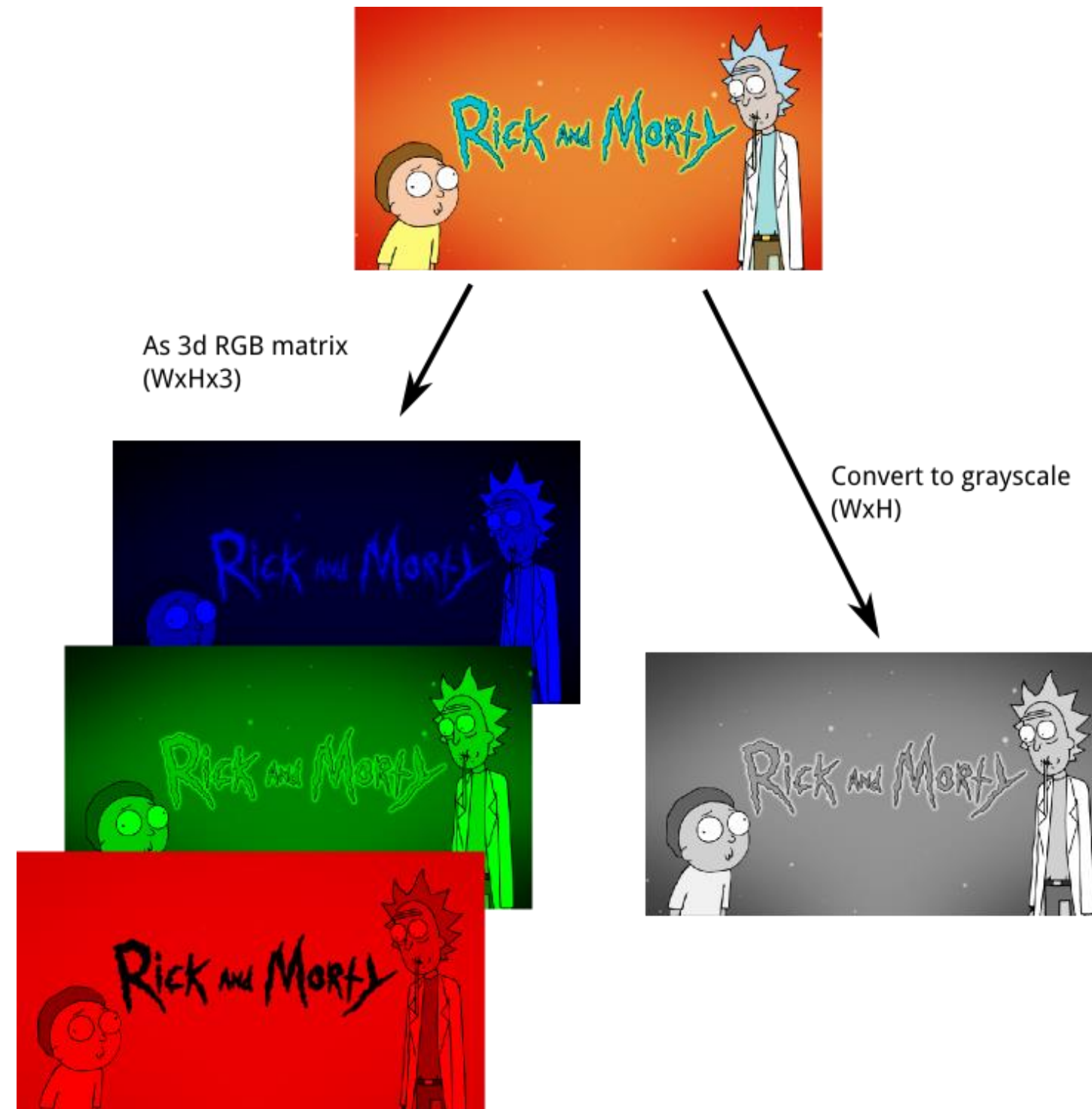
FORMATO	SIGNIFICADO	EXTENSION	DEFINICION
Portable Network Graphics	Gráfico de red portable		Es el Este formato es similar al JPEG también permite la exhibición de imágenes de amplio colorido pero su compresión no reduce la calidad de imagen como es un formato utilizado para internet posee muchas otras características.



# ESPACIOS DE COLOR

Un espacio de color es un sistema de interpretación del color, es decir, una organización específica de los colores en una imagen o video.

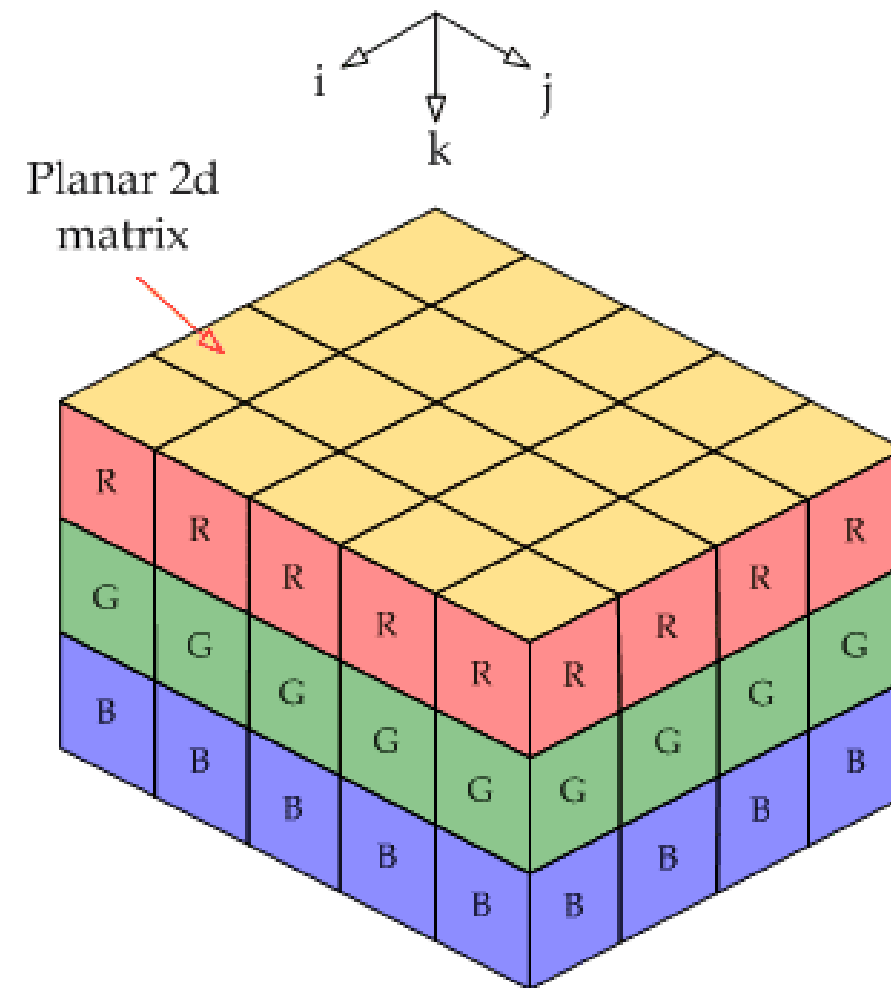
Trabajaremos  
con 2 hoy  
RGB - GRAY



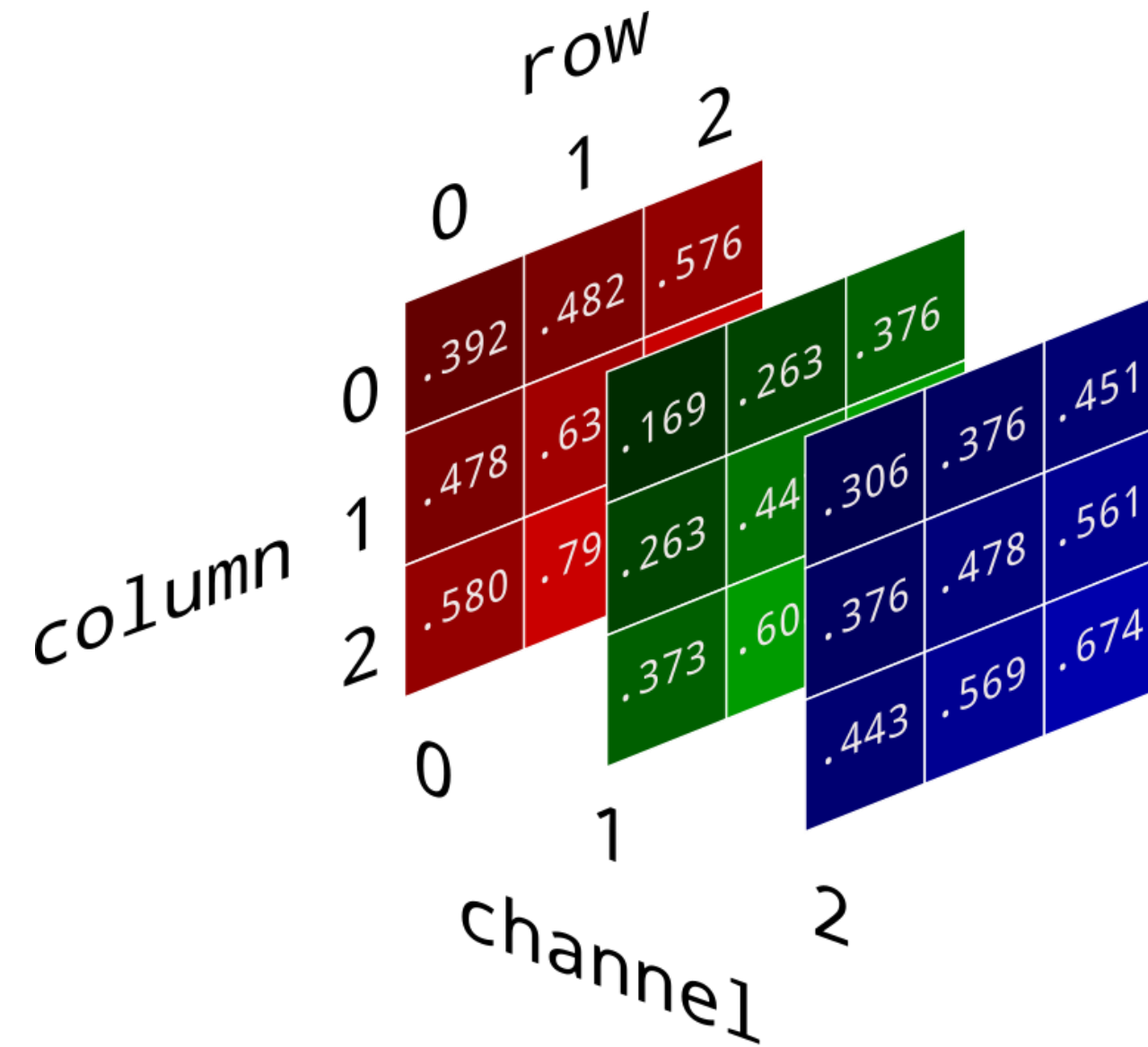
PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# ESPACIOS DE COLOR

## RGB



Graphical presentation of  
RGB 3d matrix

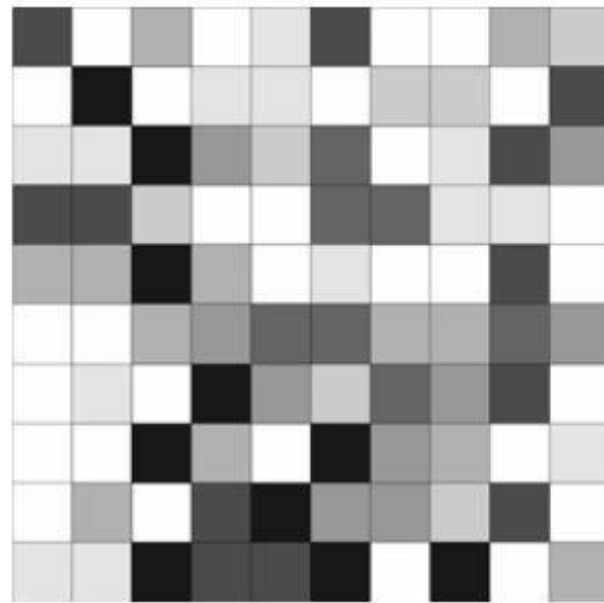


PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

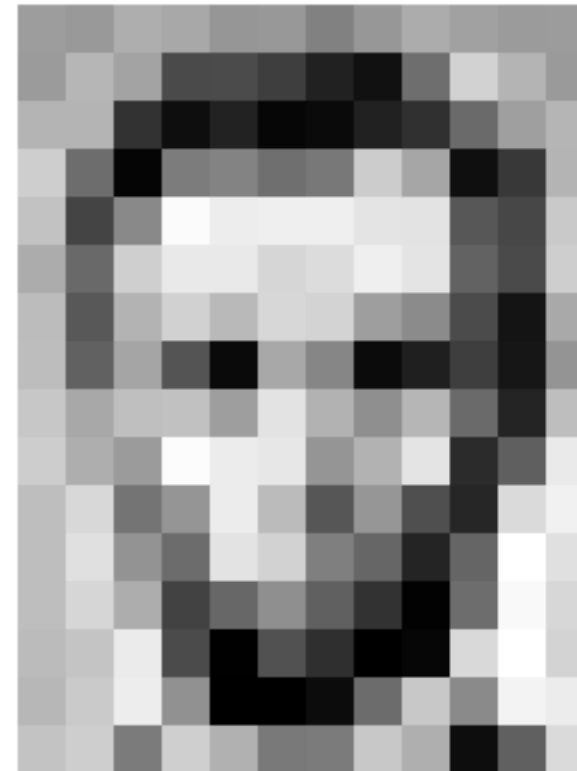


# ESPACIOS DE COLOR

## GRAY



254	107
255	165



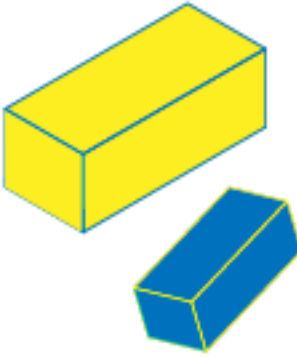
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# PROCESAMIENTO DIGITAL DE IMÁGENES (PDI) O TRATAMIENTO DE IMAGENES



Manipulación y análisis de imágenes por computadora  
Es el conjunto de técnicas que se utilizan para mejorar la apariencia visual de las imágenes

## OBJETIVOS

- Mejorar el aspecto de las imágenes
- Hacer mas evidente en ellas ciertos detalles que se desean hacer notar
- Mejorar la calidad de imagen
- Extraer características de una imagen para su descripción e interpretación



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

## Leer una imagen

Casi todas las operaciones estarán relacionadas con Numpy más que a OpenCV. Conocer bien Numpy es un requerimiento para escribir un código optimizado con OpenCV.

***Imread*** : Carga una imagen del archivo especificado. Si la imagen no se puede leer devolverá una matriz vacía

*Sintaxis:* `cv2.imread(ruta,flag)`

Hay 3 tipos de banderas:

***Cv2.IMREAD\_COLOR*** : Especifica cargar una imagen en color. Se perderá transparencia. Es la bandera predeterminada. Se puede pasar el 1

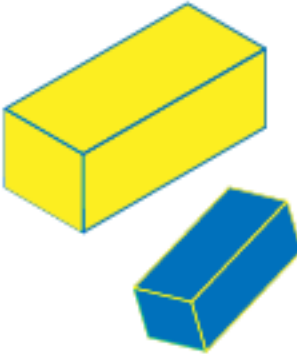
***Cv2.IMREAD\_GRAYSCALE*** : Especifica cargar una imagen en modo escala de grises.

Alternativamente se puede usar 0

***Cv2.IMREAD\_UNCHANGED*** : Especifica cargar una imagen incluyendo canal alfa. Se puede pasar el -1



## Mostrar una imagen



***imshow*** : Carga una imagen del archivo especificado. Si la imagen no se puede leer devolverá una matriz vacía

*Sintaxis:* `cv2.show(winname,imagen)`

***Parametros:***

Winname: Nombre de la ventana

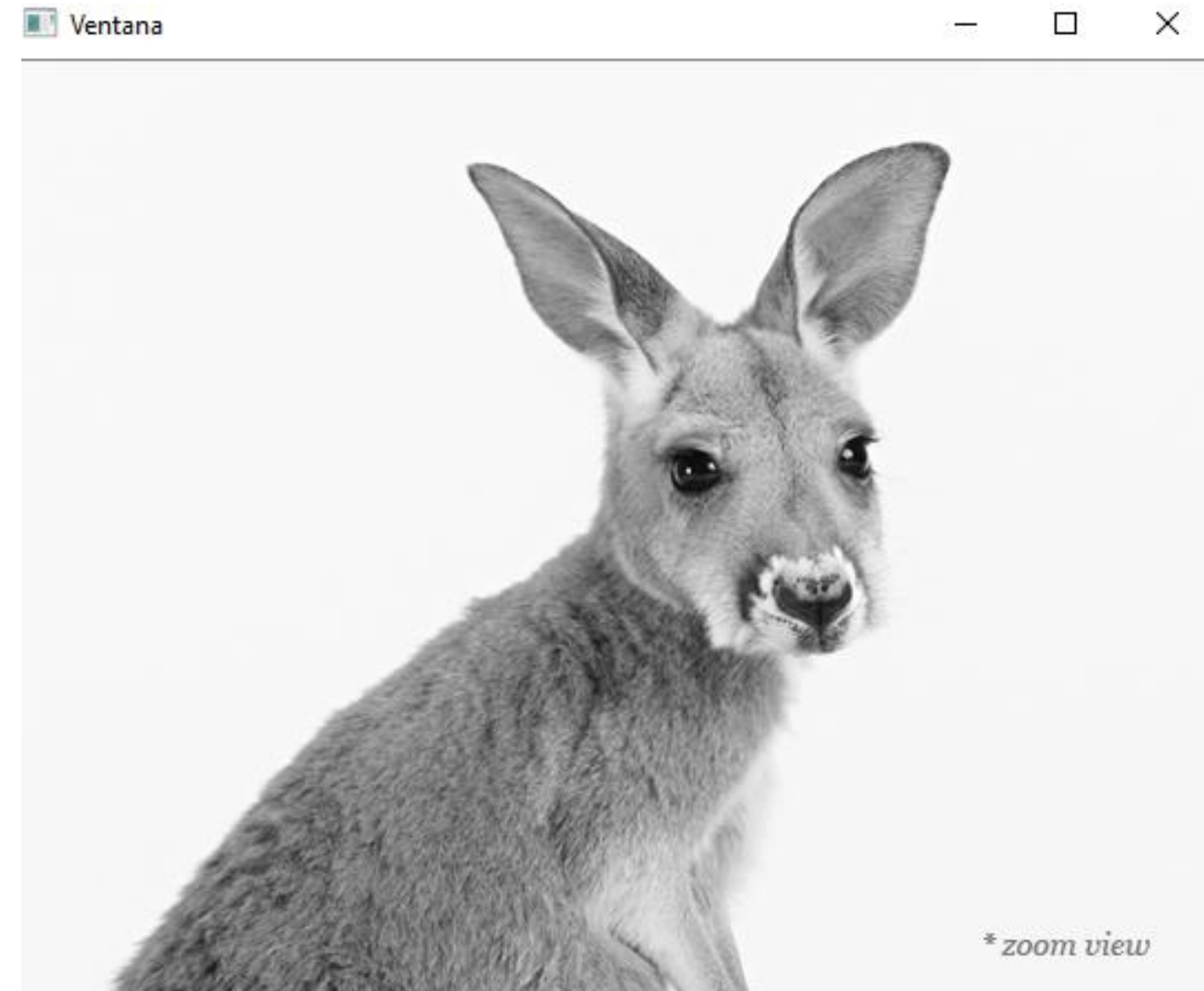
Imagen : Imagen que se mostrará

**Nota** Esta función debe ir seguida de la ***waitKey*** función que muestra la imagen durante milisegundos especificados. De lo contrario, no mostrará la imagen. Por ejemplo, ***waitKey(0)*** mostrará la ventana infinitamente hasta que se presione cualquier tecla (es adecuada para la visualización de imágenes). ***waitKey(25)*** mostrará un cuadro durante 25 ms, después de lo cual la pantalla se cerrará automáticamente. (Si lo coloca en un bucle para leer videos, mostrará el video cuadro por cuadro).



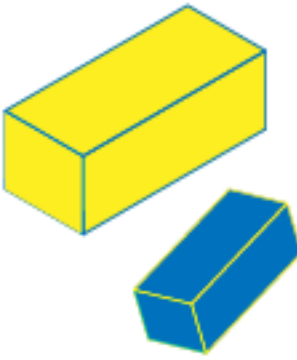
## Ejemplo 1:

```
import cv2  
  
img = cv2.imread("img.jpg",0)  
cv2.imshow("Ventana",img)  
  
cv2.waitKey()
```



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# Guardar una imagen



***imwrite*** : Guarda una imagen del archivo especificado.

*Sintaxis:* `cv2.show(filename,img,params)`

***Parametros:***

filename: Imagen o imágenes a guardar

Img : Imagen que se mostrará

Params : Formatos de codificación



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

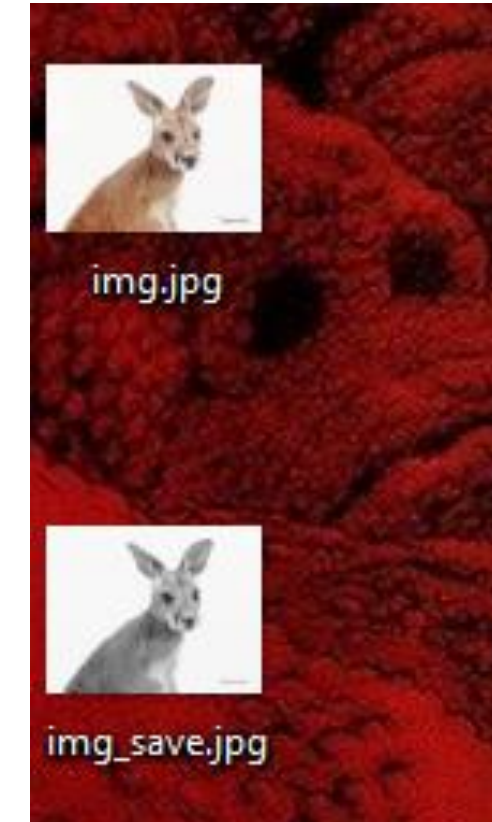


## Resumiendo

```
import cv2

img = cv2.imread("img.jpg",0)
cv2.imshow("Ventana",img)

if cv2.waitKey(0) ==27: #tecla esc para salir
    cv2.destroyAllWindows()
if cv2.waitKey(0) == ord('s'): # tecla s
    cv2.imwrite("img_save.jpg",img)
    cv2.destroyAllWindows()
```



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# Usando Matplotlib

Matplotlib es una biblioteca para graficar en Python que te ofrece una amplia variedad de métodos para plotear. Se puede ampliar las imágenes, guardarlas, etc

```
import numpy as numpy
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("img.jpg",0)
plt.imshow(img)
plt.xticks([])
plt.yticks([])
plt.show()
```

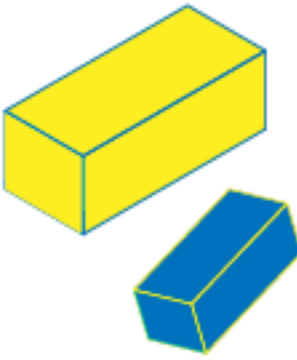


La imagen a color cargada por OpenCV está en modo BGR. Pero Matplotlib se muestra en modo RGB. Por lo tanto, las imágenes a color no se mostrarán correctamente en Matplotlib si se lee la imagen con OpenCV. s



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

# Empezando con videos



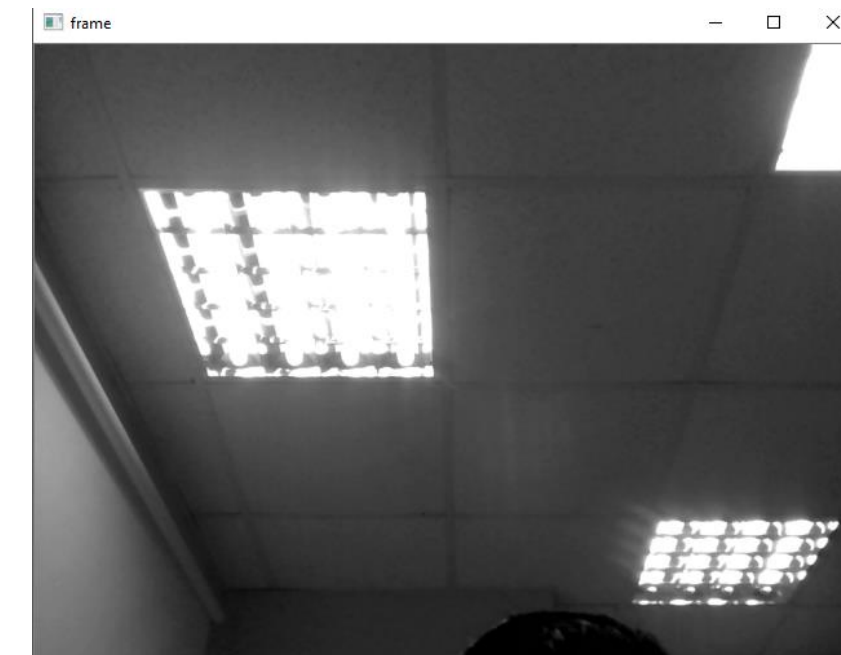
## Captura de vídeo desde la cámara

A menudo, tenemos que capturar transmisiones en vivo con la cámara. OpenCV proporciona una interfaz muy simple para esto. Vamos a capturar un vídeo desde la cámara (estoy utilizando una webcam incorporada de mi pc), convirtiendo en vídeo a escala de grises y mostrándolo. Es una tarea sencilla para empezar.

Para capturar un vídeo, es necesario crear un objeto VideoCapture. Su argumento puede ser el índice del dispositivo o el nombre de un archivo de vídeo. El índice del dispositivo es el número para especificar qué cámara se utilizará. Normalmente se conectará una cámara (como en mi caso). Así que simplemente paso 0 (o -1). Puedes seleccionar una segunda cámara pasando 1 y así sucesivamente. Después de esto, puedes capturar el vídeo. Pero al final, no te olvide de liberar la captura.

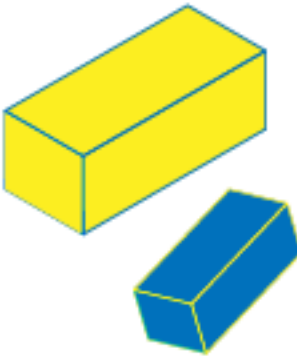
```
import cv2
cap = cv2.VideoCapture(0)

while True:
    #Captura video cuadro a cuadro
    ret , frame =cap.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    #muestra cuadro resultante
    cv2.imshow("ventana",gray)
    if cv2.waitKey(1)&0xFF == ord("s"):
        break
#cuando todo esta listo, se libera la captura
cap.release()
cv2.destroyAllWindows()
```



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES

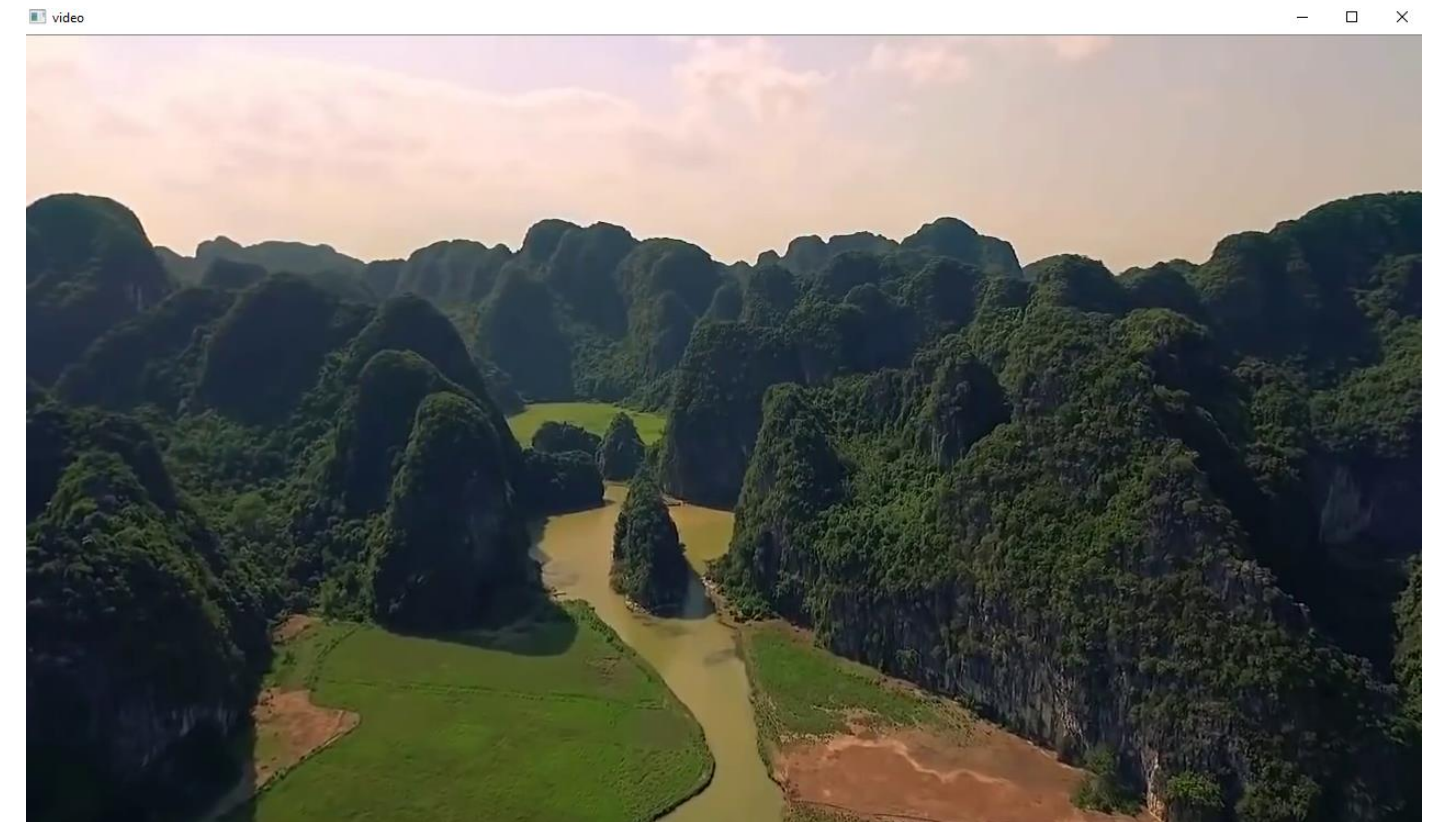
# Empezando con videos



## Reproducción de video desde un archivo

Es lo mismo que capturar desde la cámara , solo se cambia el índice de la cámara con el nombre del archivo de video. También mientras que se visualice el cuadro, utiliza el tiempo apropiado para `cv2.waitKey()`. Si es demasiado corto, el video será muy rápido y si es demasiado alto, el video será lento (Una forma de mostrar videos en cámara lenta). 25 milisegundos estará bien para casos normales

```
import cv2
captura = cv2.VideoCapture('vi.mp4')
while (captura.isOpened()):
    ret, imagen = captura.read()
    if ret == True:
        cv2.imshow('video', imagen)
        if cv2.waitKey(30) == ord('s'):
            break
    else: break
captura.release()
cv2.destroyAllWindows()
```



PROCESAMIENTO  
DIGITAL DE  
IMÁGENES



# Empezando con videos

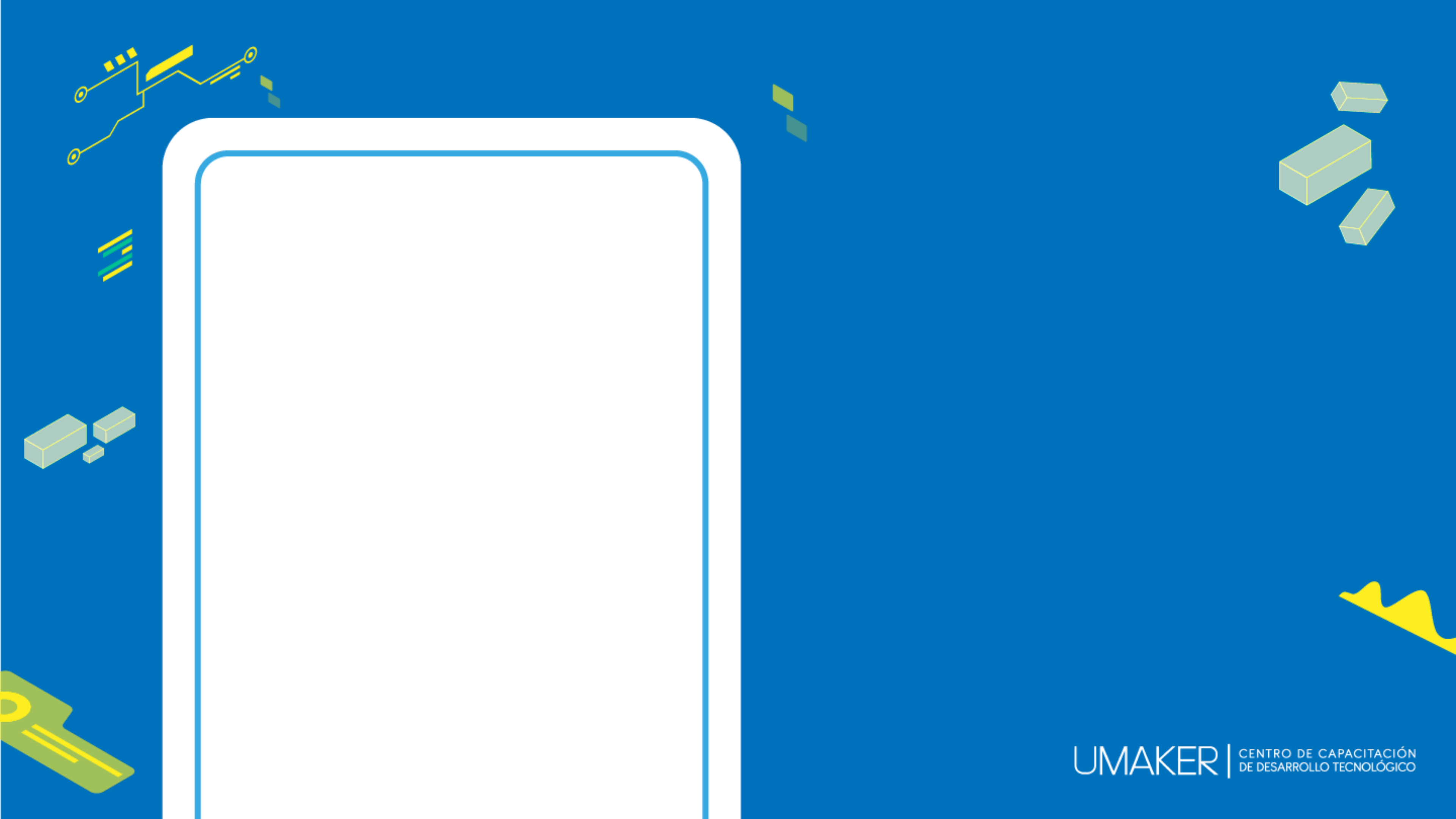
## Guardando un vídeo

Supongamos que queremos guardar un video que hemos previamente capturado y procesado cuadro por cuadro. Para el caso de las imágenes, esto es muy simple, solo necesitamos usar `cv2.imwrite()`. En el caso de los videos necesitamos crear un objeto `VideoWriter()`. Primero tenemos que especificar el nombre que le queremos dar, Luego el códec del video, cuadros por segundo y tamaño de los fotogramas (640,480).

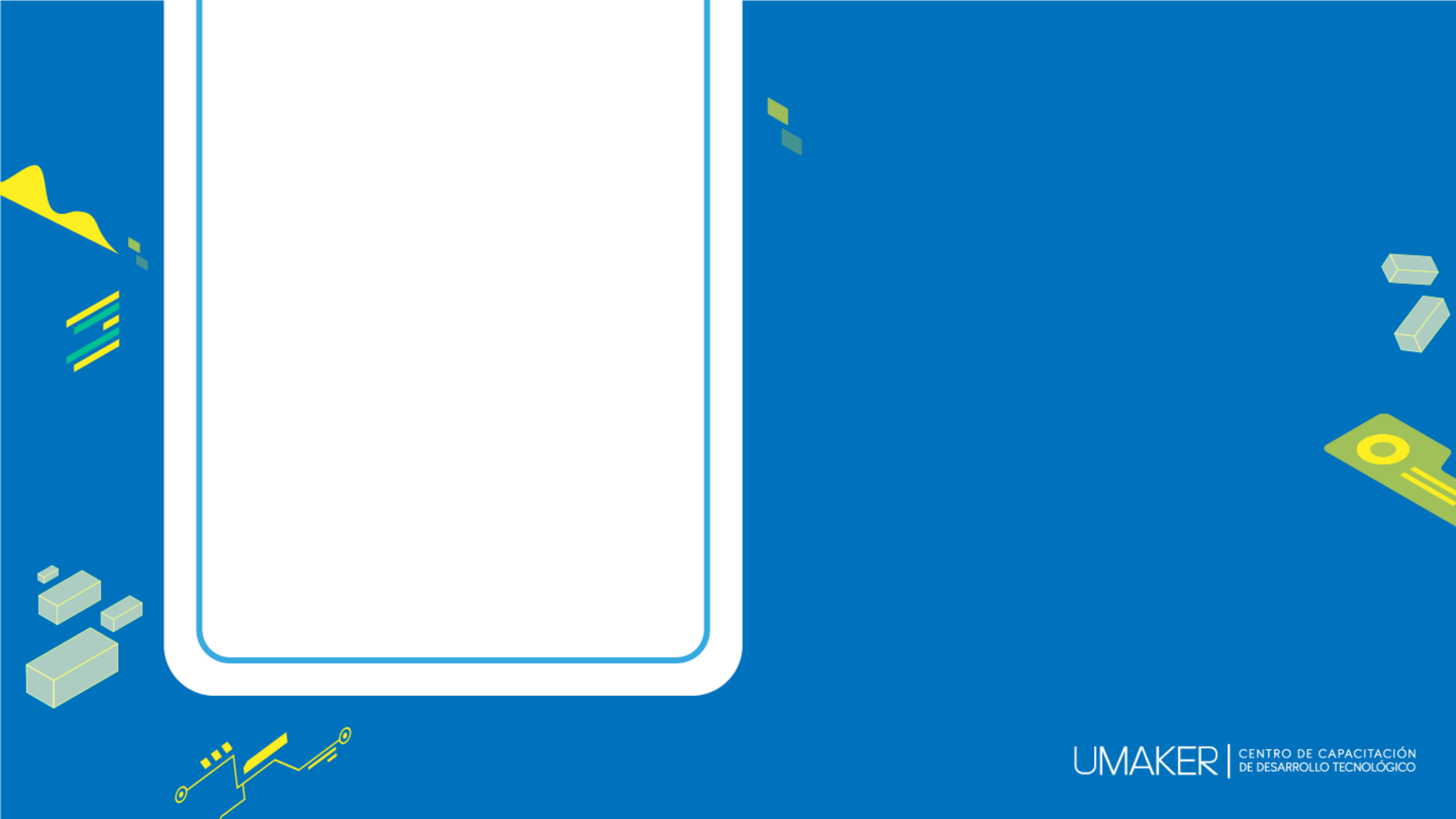
La función `write()` especificamos la imagen o fotograma que queremos que se almacene.

```
import cv2
captura = cv2.VideoCapture(0)
salida = cv2.VideoWriter('vi_salida.avi',cv2.VideoWriter_fourcc(*'DIVX'),20.0,(640,480))
while (captura.isOpened()):
    ret, imagen = captura.read()
    if ret == True:
        cv2.imshow('video', imagen)
        salida.write(imagen)
        if cv2.waitKey(1) & 0xFF == ord('s'):
            break
    else: break
captura.release()
salida.release()
cv2.destroyAllWindows()
```









UMAKER | CENTRO DE CAPACITACIÓN  
DE DESARROLLO TECNOLÓGICO