

# INTRODUCCIÓN

M. en I. Miguel Angel Camargo  
Rojas

FACULTAD DE INGENIERÍA

UNIVERSIDAD PANAMERICANA

AGOSTO - DICIEMBRE, 2023



# CONTENIDO



**LAS ONDAS**



**VISIÓN HUMANA**



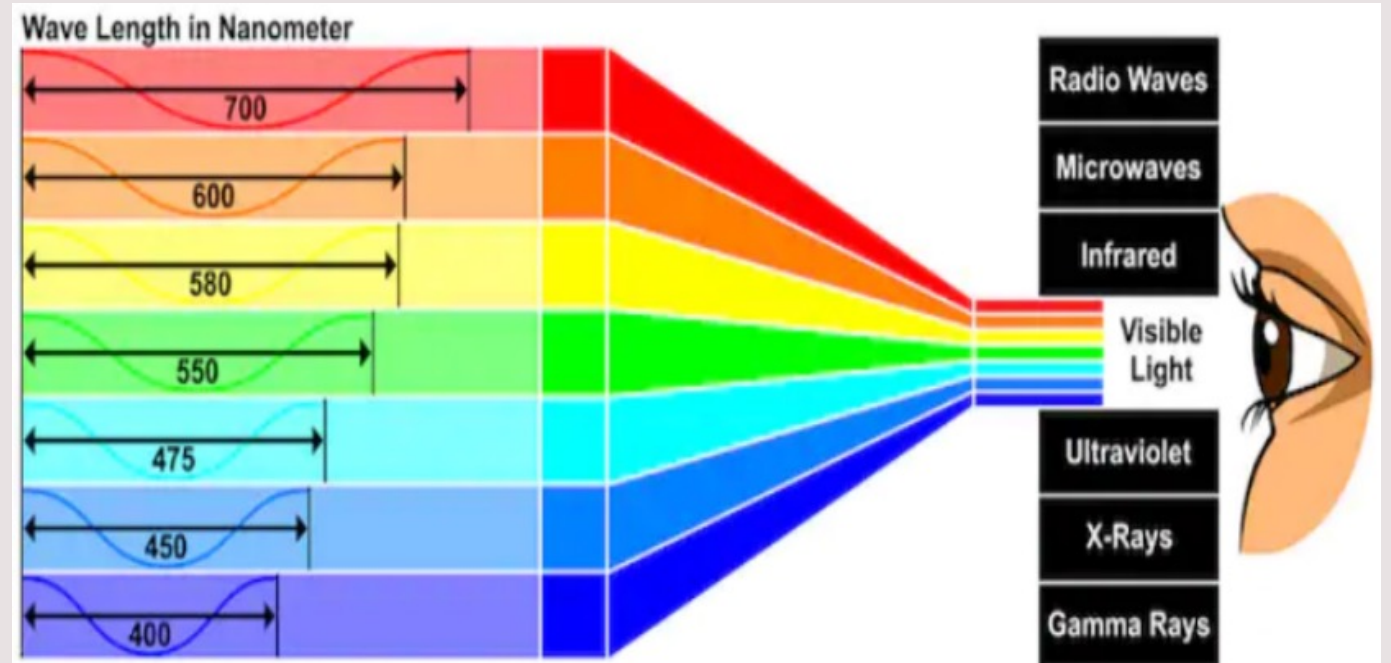
**VERSIÓN DIGITAL DE UNA IMAGEN**



**PYTHON**

# LAS ONDAS

- La luz es radiación electromagnética (se puede propagar en el vacío) que viaja en ondas.
- El sonido es una onda mecánica.



*Espectro visible*



## Todas las ondas tienen frecuencia y amplitud

- En el sonido la frecuencia es el tono, mientras que la amplitud es el volumen.
- En la luz la frecuencia → tono, mientras que la amplitud → brillo, es decir a mayor amplitud, mayor brillo.
- ¿Qué colores tienen alta frecuencia, y cuáles baja?
- ¿En la figura anterior, se puede observar variaciones en amplitud?

# VISIÓN HUMANA

## FUN FACTS

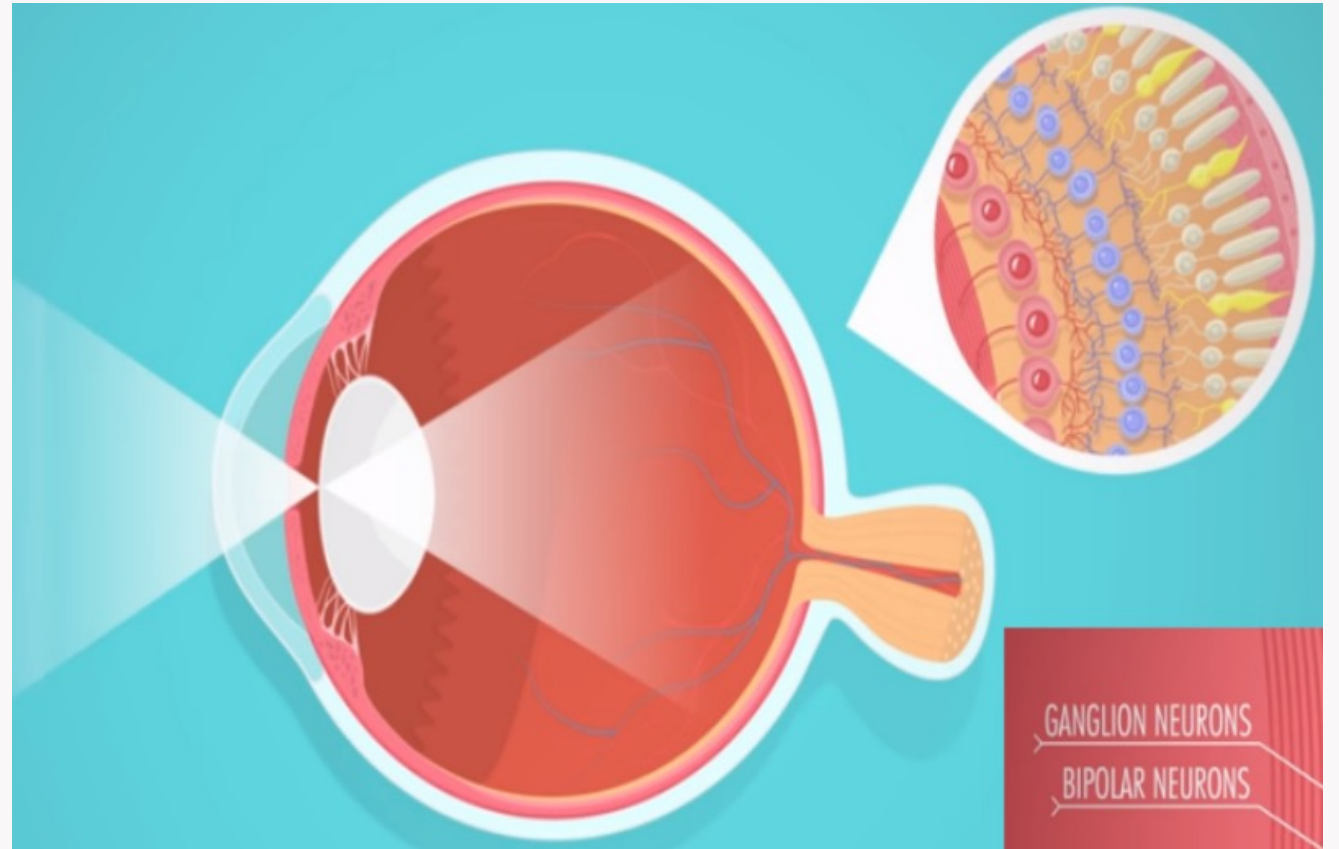
- 70% de todos nuestros receptores sensoriales se emplean en nuestra visión.
- La vista es considerada como nuestro sentido dominante
- Nuestra resolución es de aproximadamente 576 MP.
- El globo ocular mide aprox. 2.5 cm.

# ANATOMÍA OCULAR

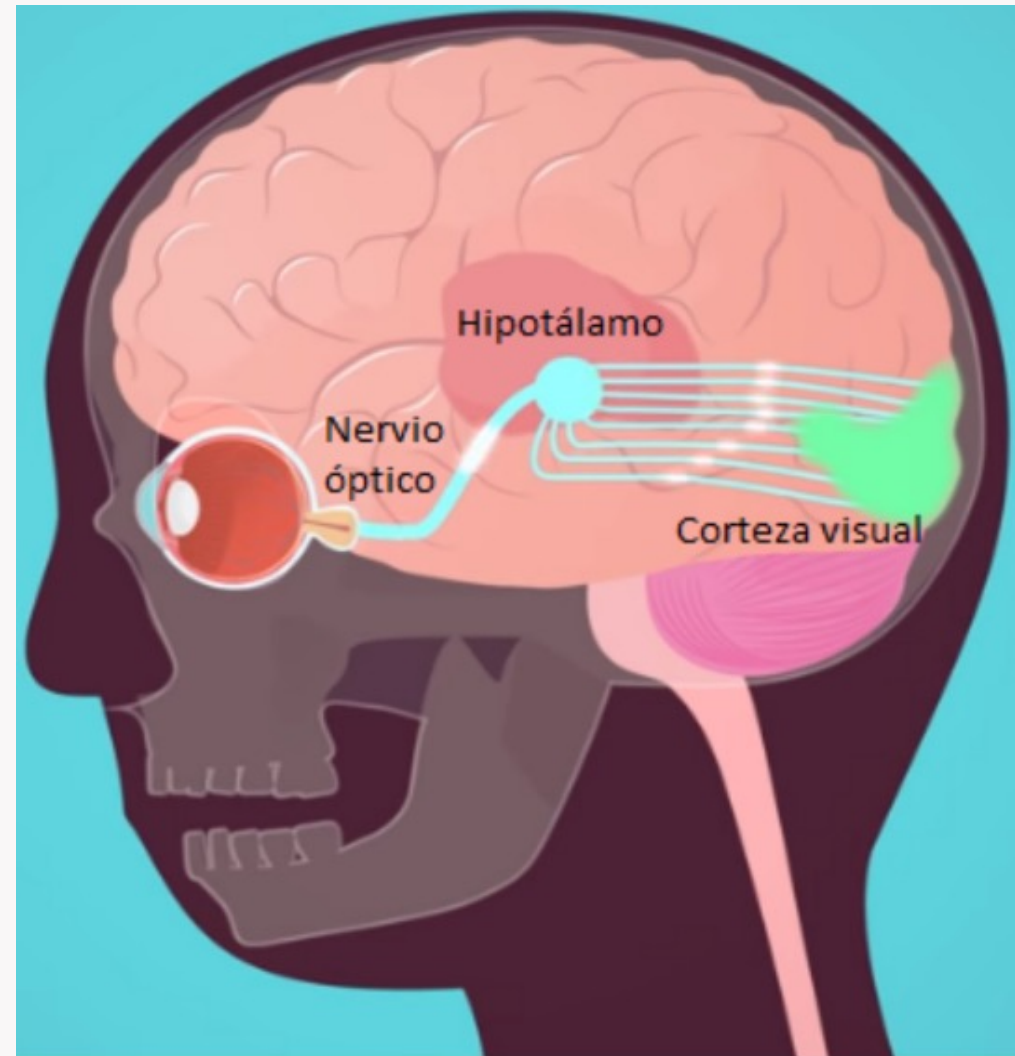




# ANATOMÍA OCULAR: NEURONAS Y FOTO SENSORES

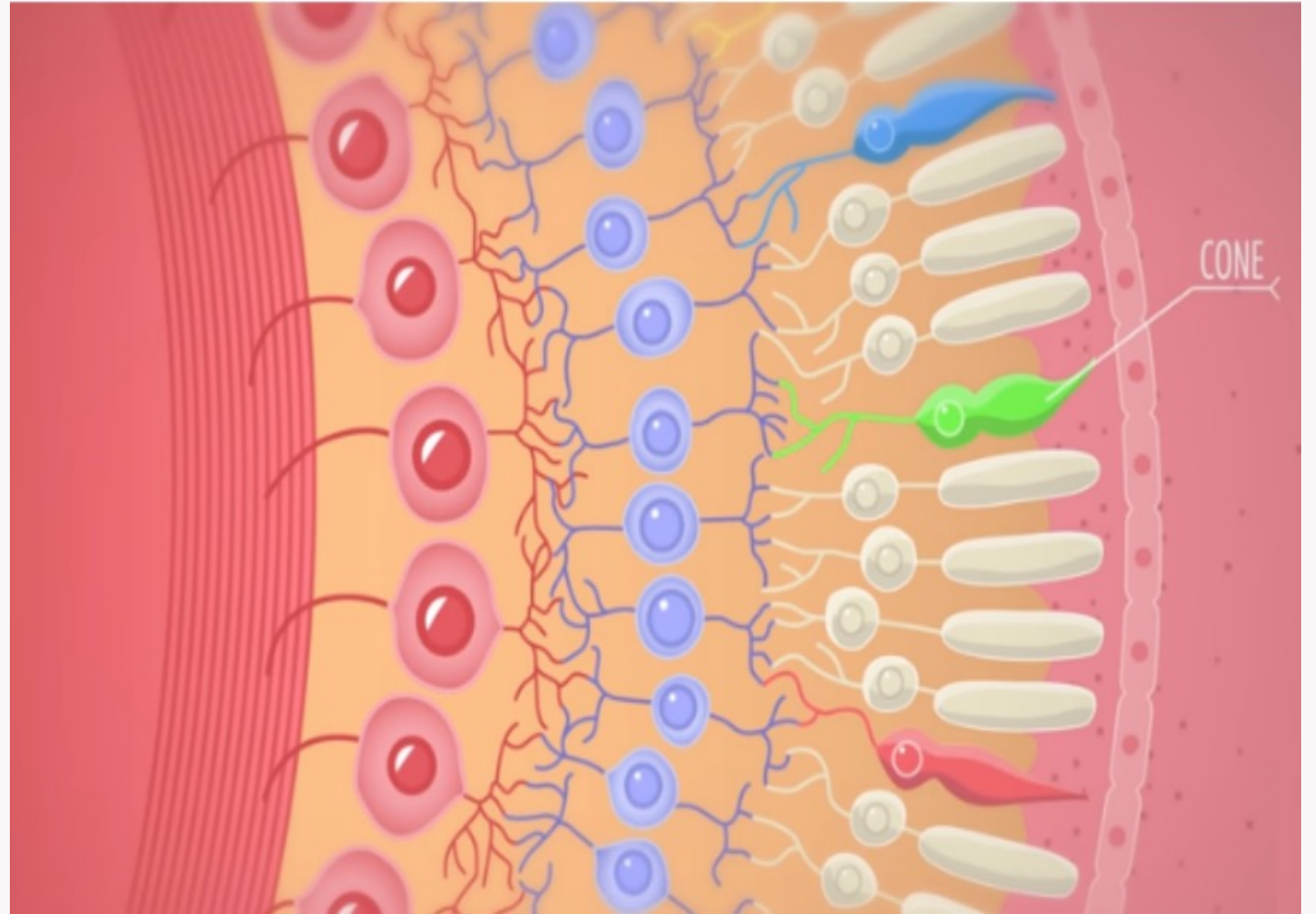


# ANATOMÍA OCULAR

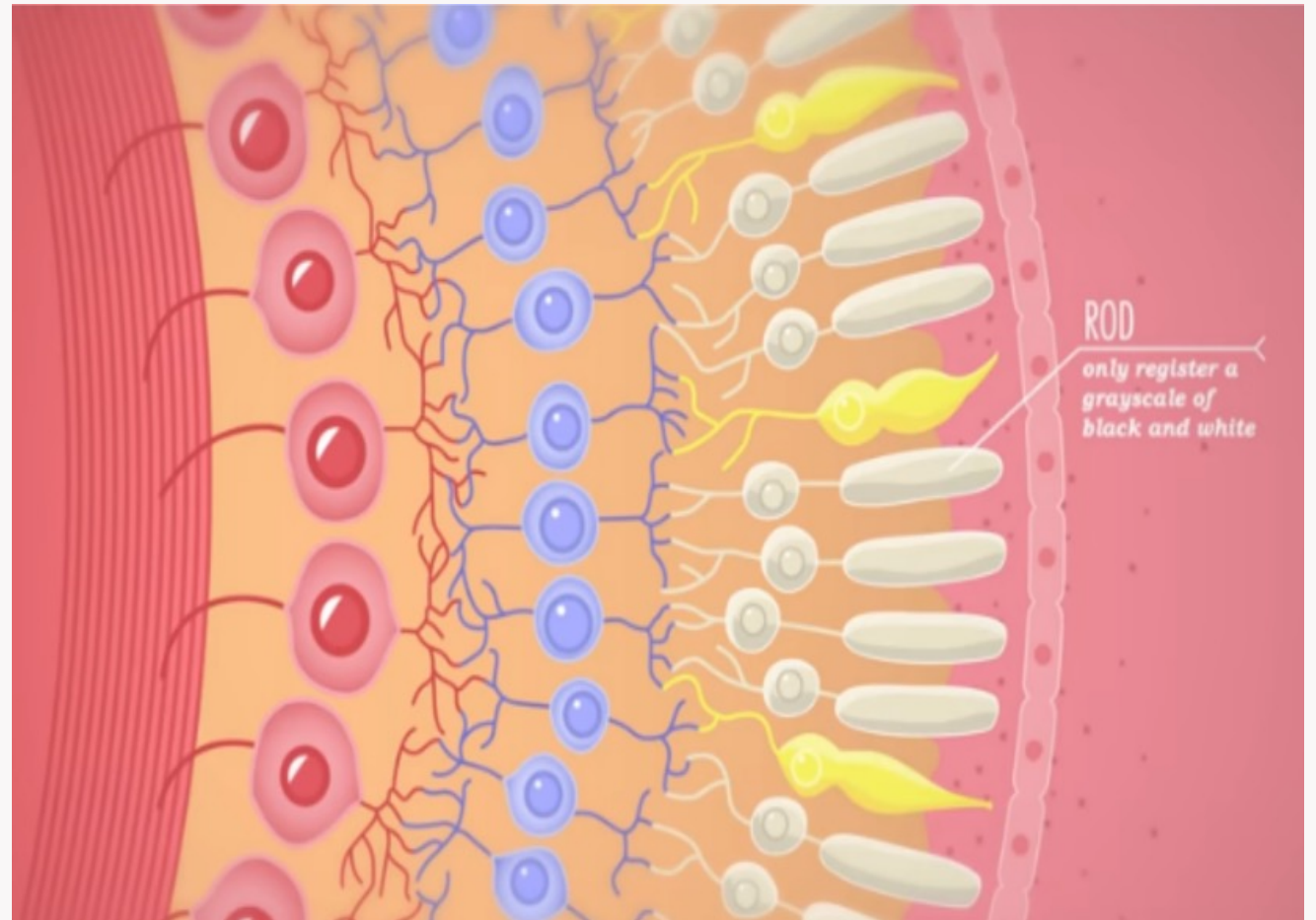




# ANATOMÍA OCULAR: CONOS PARA EL COLOR



# ANATOMÍA OCULAR: BASTONES (ESCALA DE GRISES)



# CONOS Y BASTONES

## Conos

- Detalles finos,
- Colores: rojo, verde y azul,
- Trabajan optimamente en presencia de luz.



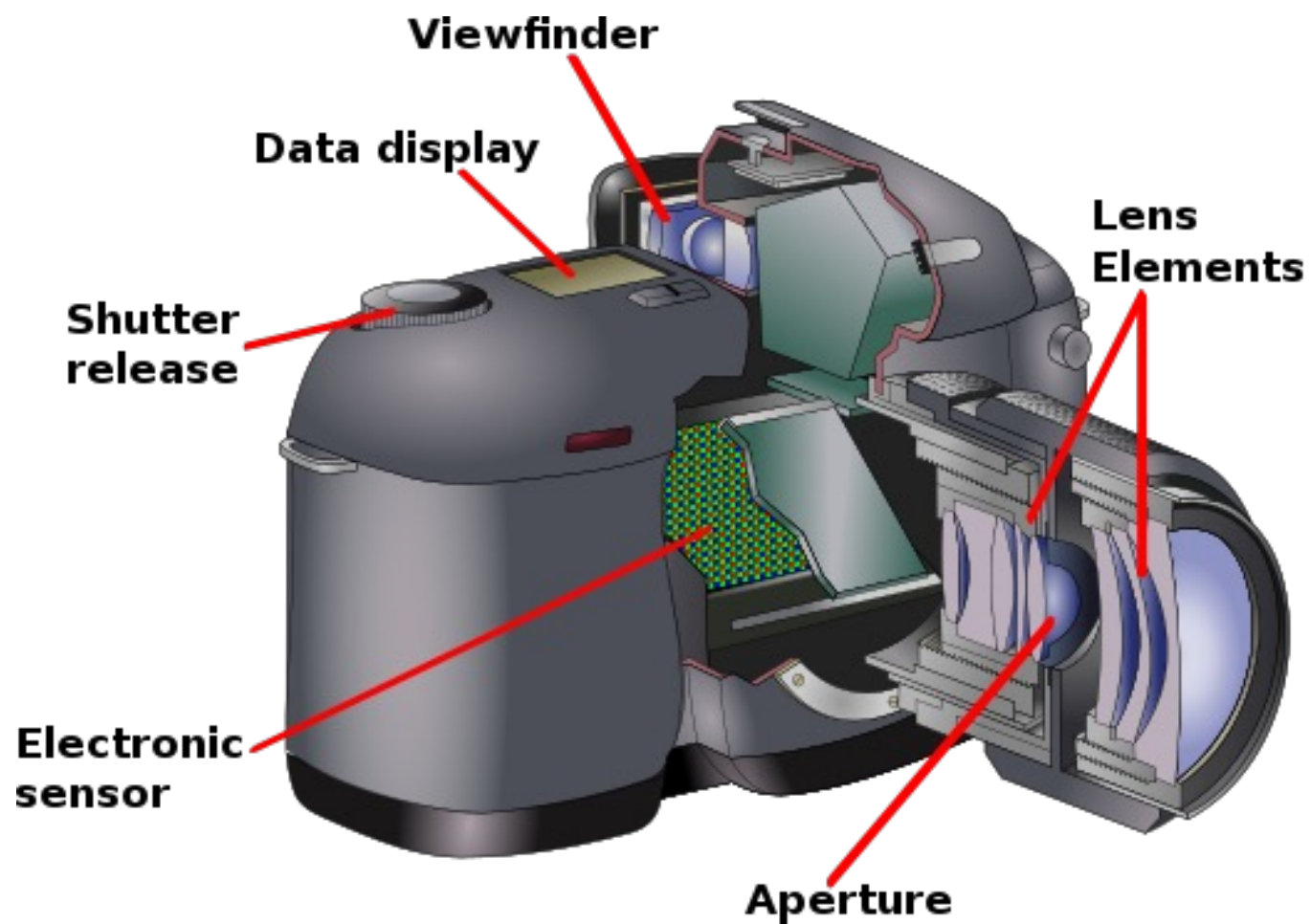
## Bastones


- Tienden a aglomerarse,
- Registran formas generales,
- Escala de grises,
- Ausencia de luz.



# CÁMARA DIGITAL





The background of the image is a dark, artistic photograph of a camera lens. It shows the internal elements of the lens, including the aperture blades which are partially open, creating a hexagonal shape. Light rays are visible, entering from the left and reflecting off the lens elements, creating a bokeh effect with soft, out-of-focus light spots in shades of blue, green, and yellow. The overall tone is dark and moody.

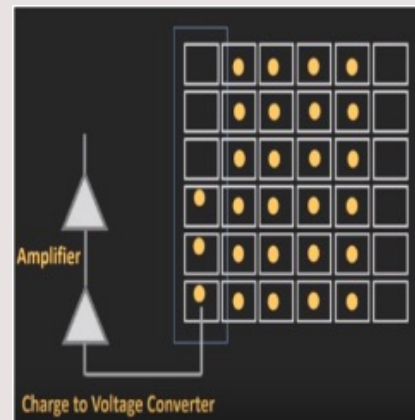
# EL SENSOR DE UNA CÁMARA



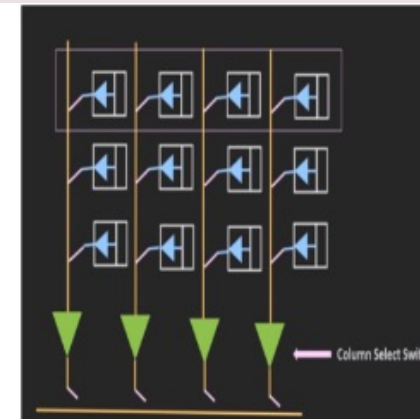
# SENSORES

¿Qué tipos de sensores hay?

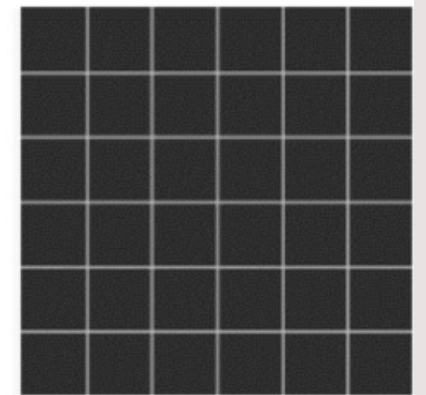
- CMOS: Complementary Metal Oxide Semiconductor
- CCD: Charge-Coupled Decive.
- Ambos tienen como objetivo hacer la conversión A/D



(a) CCD



(b) CMOS

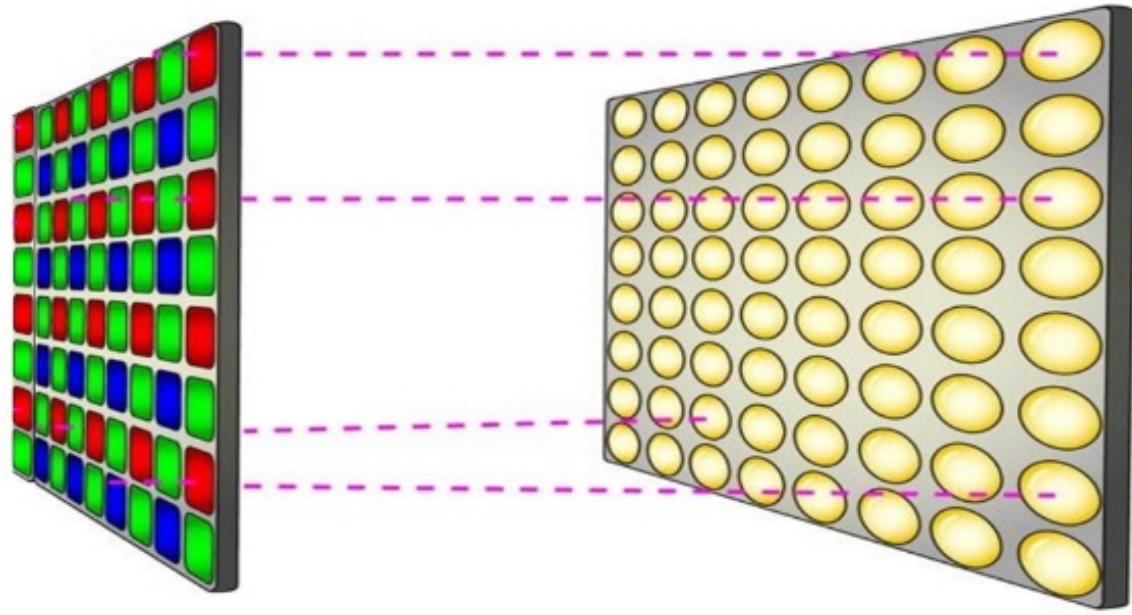


(c) Arreglo de un sensor

*Sensores*

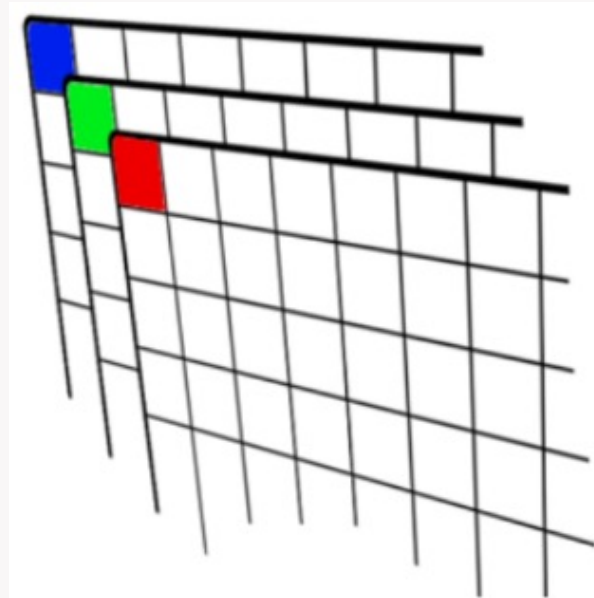
## Distribución

- Green → 50%
- Red → 25%
- Blue → 25%

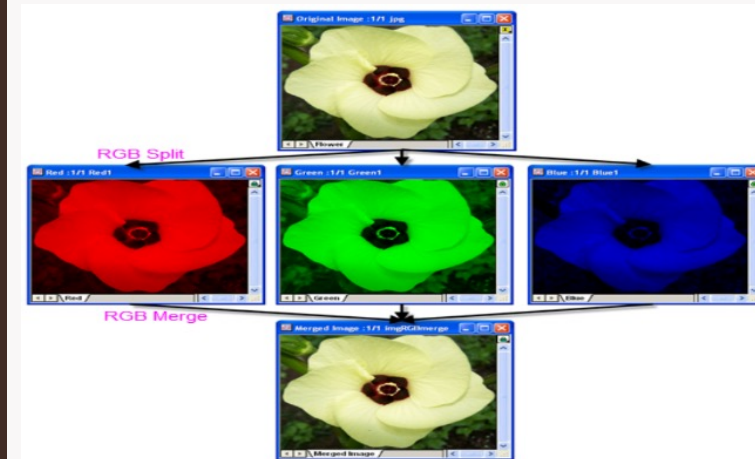


# FILTRO DE BAYER

# FILTRO DE BAYER

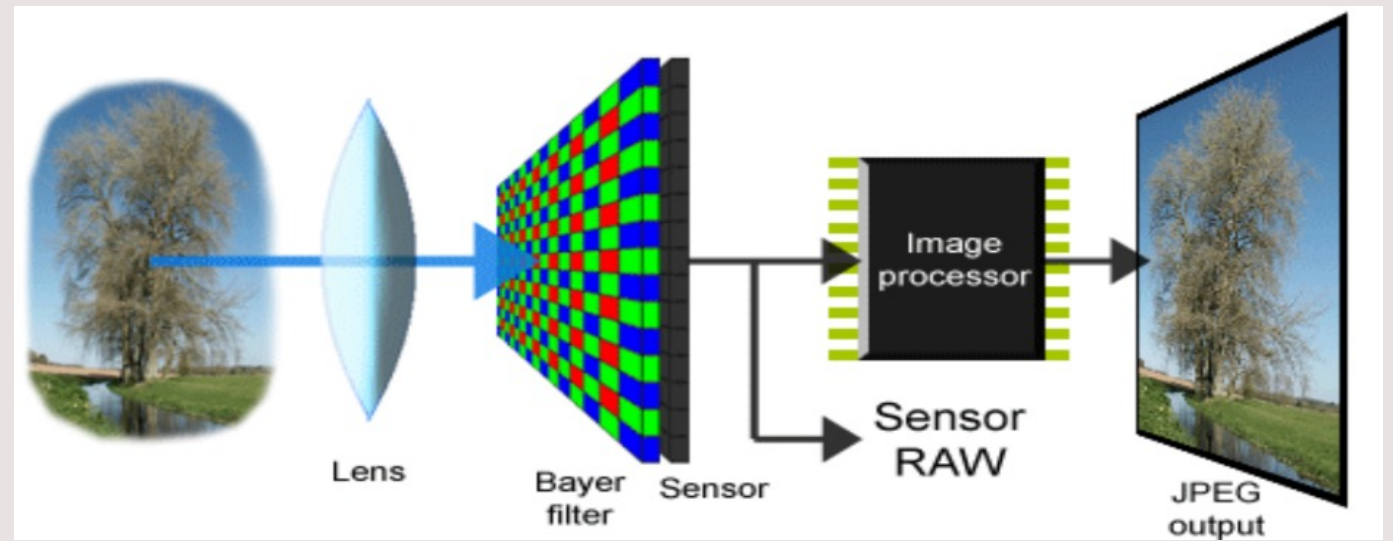


*RGB*

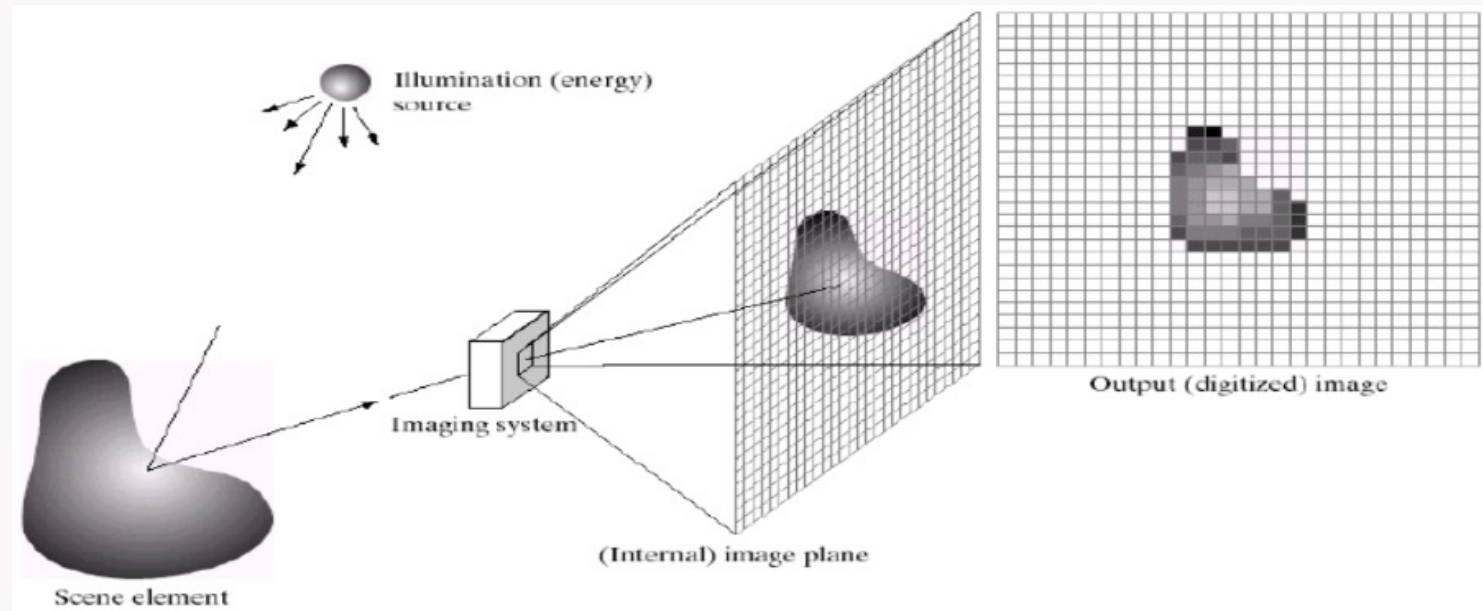


*RGB CHANNELS*

*La cámara fotográfica, el  
filtro de Bayer y el sensor*



UNA IMAGEN DIGITAL ES  
UNA REPRESENTACIÓN DE 2  
DIMENSIONES, LA CUAL  
ESTÁ COMPUESTA DE  $M \times N$   
PICTURE ELEMENTS O  
PIXELES.





# PYTHON

Python es un lenguaje de programación de alto nivel interpretado, orientado a objetos y con tipado dinámico.

La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo de mantenimiento del programa. Python admite módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización del código.





# OpenCV

OpenCV (Open Source Computer Vision Library: <http://opencv.org>) es una biblioteca de código abierto que incluye varios cientos de algoritmos de visión artificial. La documentación describe la llamada API OpenCV 2.x, que es esencialmente una API hecha en C++, a diferencia de la API OpenCV 1.x basada en C (la API C está obsoleta y no se prueba con el compilador "C" desde las versiones OpenCV 2.4)

# OpenCV-Python

OpenCV-Python es una biblioteca de enlaces de Python diseñada para resolver problemas de visión por computadora.

Comparado con lenguajes como C/C++, Python es más lento. Dicho esto, Python se puede ampliar fácilmente con C/C++, lo que nos permite escribir código computacionalmente intensivo en C/C++ y crear contenedores de Python que se pueden usar como módulos de Python. Esto nos da dos ventajas: primero, el código es tan rápido como el código C/C++ original (ya que, de hecho, es el código C++ trabajando en segundo plano) y segundo, es más fácil codificar en Python que en C/C++. OpenCV-Python es un contenedor de Python para la implementación original de OpenCV C++.

# OpenCV-Python

OpenCV-Python hace uso de Numpy, que es una biblioteca altamente optimizada para operaciones numéricas con una sintaxis de estilo MATLAB. Todas las estructuras de arreglos de OpenCV se convierten en arreglos Numpy. Esto también facilita la integración con otras bibliotecas que usan Numpy, como SciPy y Matplotlib.

# BASICS

- ¿Cómo leer una imagen?
- Canales RGB
- Tamaño de mi imagen
- Acceder a un pixel en particular
- Desplegar (mostrar) mi imagen

- ¿Cómo leer una imagen?
- Canales RGB

```
# Leer imagen a color
image = cv.imread('./images/Piano_Sheet_Music.png')

# Canales BGR
canal_B = image[:, :, 0]
canal_G = image[:, :, 1]
canal_R = image[:, :, 2]
```

- Tamaño de mi imagen

```
# Tamaño de mi imagen
print(image.shape)
```

```
(854, 1132, 3)
```

- Acceder a un pixel en particular  
 $f(i,j,ch)$  donde  $i$  = número de fila,  $j$  = número de columna y  $ch$  = número de canal

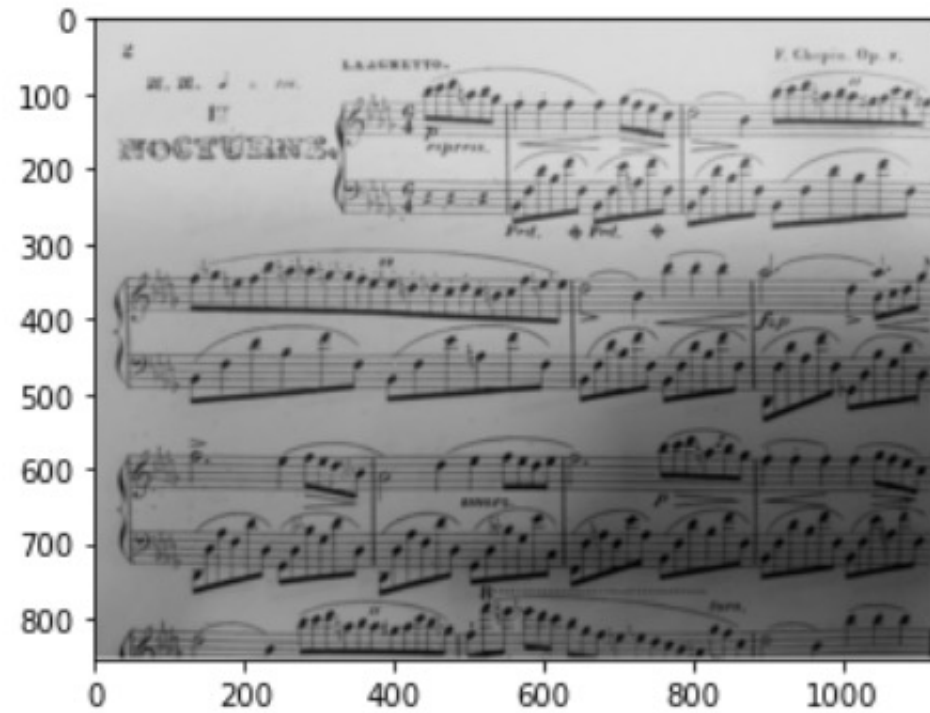
```
# Acceder a un pixel en particular
image[800,800,1]
```

```
98
```

# DESPLEGAR IMAGEN

```
plt.imshow(image)
```

```
<matplotlib.image.AxesImage at 0x1eada27d4c0>
```







GRACIAS POR  
SU ATENCIÓN

FIN