



De sterrennacht, Vincent van Gogh

CC3501 Colores

Eduardo Graells-Garrido
egraells@dcc.uchile.cl
Otoño 2023



[https://commons.wikimedia.org/wiki/File:Anna's_Hummingbird_\(m\)_\(39583395995\).jpg](https://commons.wikimedia.org/wiki/File:Anna's_Hummingbird_(m)_(39583395995).jpg)



Fuente: CMU



on screen



printed

¿A quién no le ha pasado?

¿Qué es el color?

Color

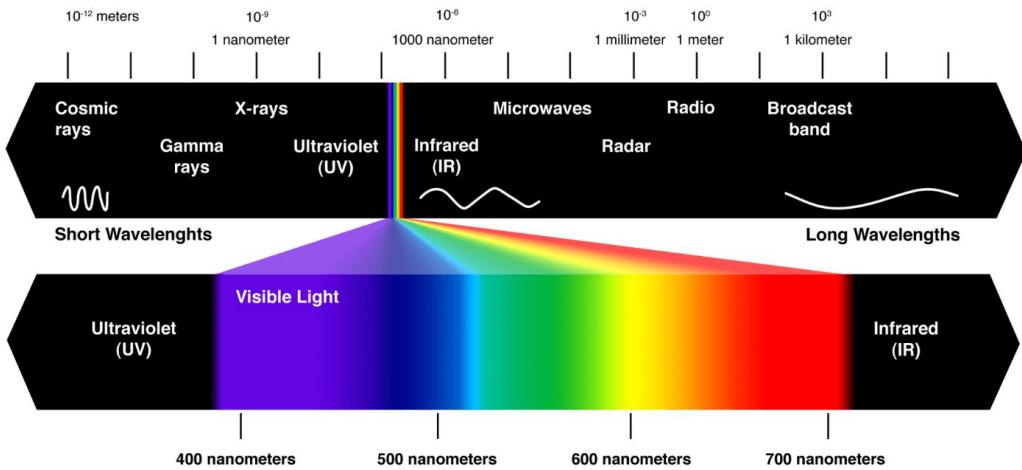
La luz es radiación electromagnética oscilante.

El color es su frecuencia.

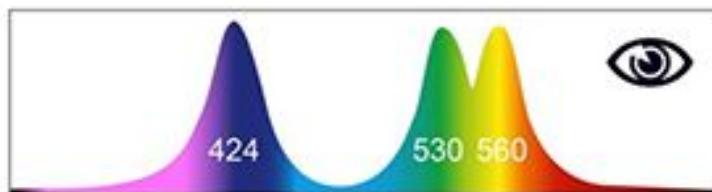
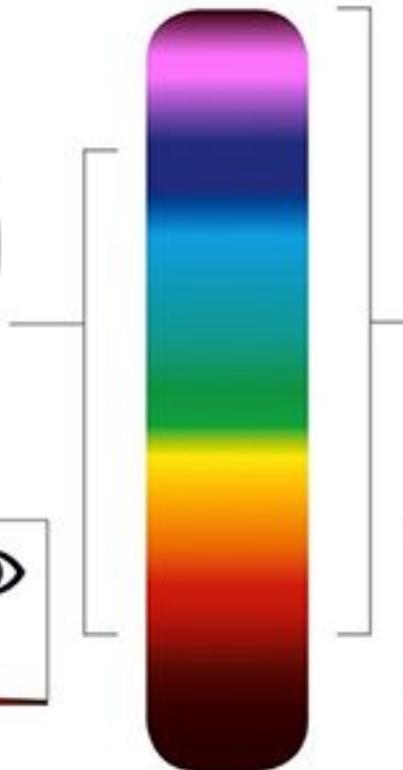
El espectro visible humano se sitúa entre 400nm (azul) y 700nm (rojo).

Máquinas pueden ver más que eso: rayos X, infrarrojo, ondas de radio.

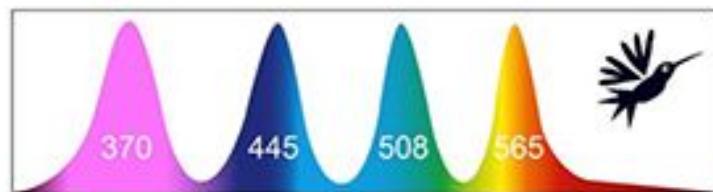
¡Las aves pueden ver luz ultravioleta!



Human Vision



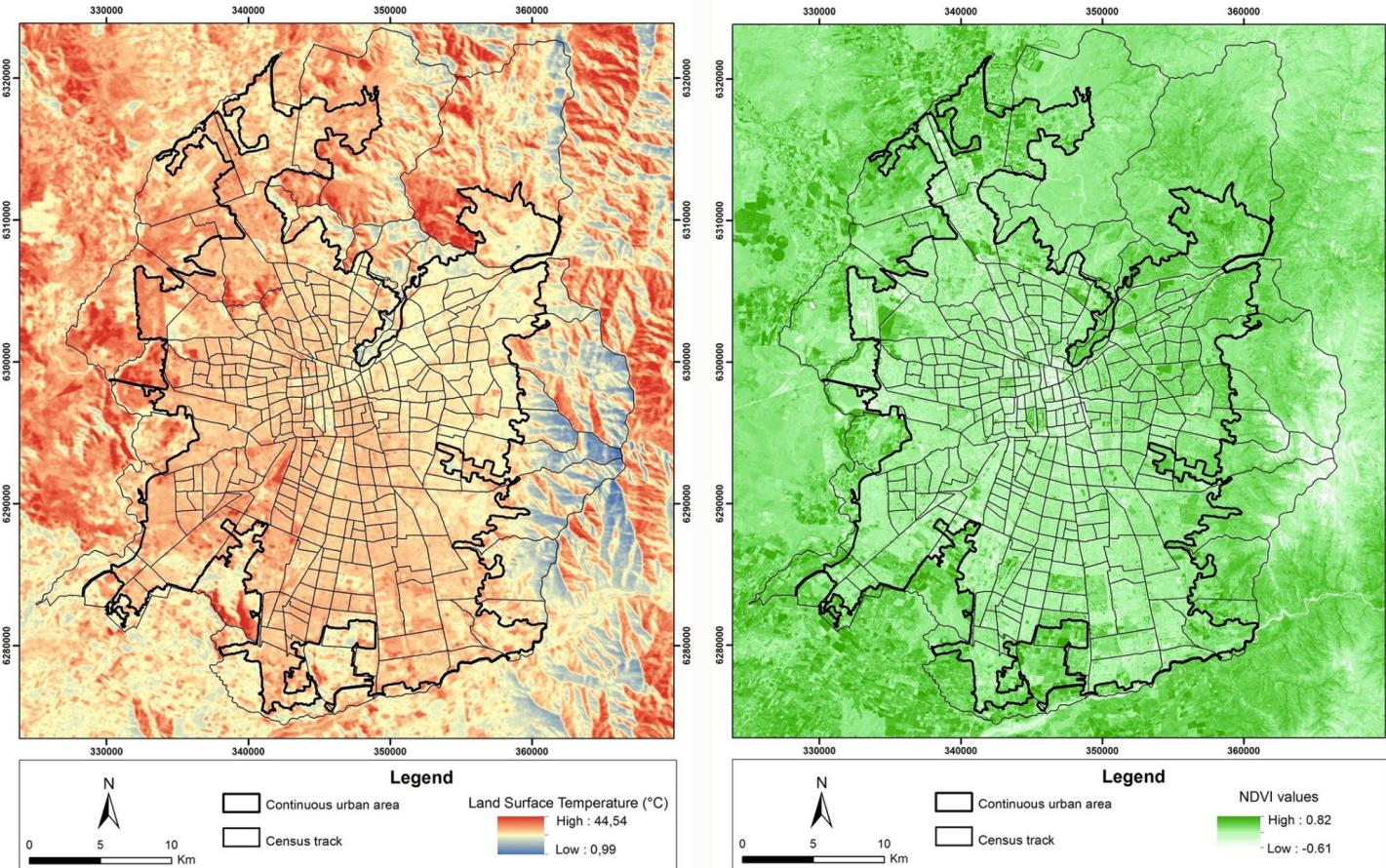
Bird Vision





Fuente: <https://www.boredpanda.com/human-vs-bird-vision>

:Las bandas que no observamos se pueden estudiar! Por ej., en las imágenes satelitales se utilizan para determinar la temperatura y el índice de vegetación de un lugar (de un pixel de la imagen).





El color no es una propiedad estática de los objetos. ¿Por qué se pone rojo el neón cuando se calienta?

El calor genera luz

Una de las tantas maneras en que se produce luz es el calor.

Maxwell: el movimiento de partículas crea un campo electromagnético.

Termodinámica: las partículas se mueven constantemente.

¡Lo que se mueve, genera luz!

Como implicancia, cada objeto alrededor de ustedes produce colores.

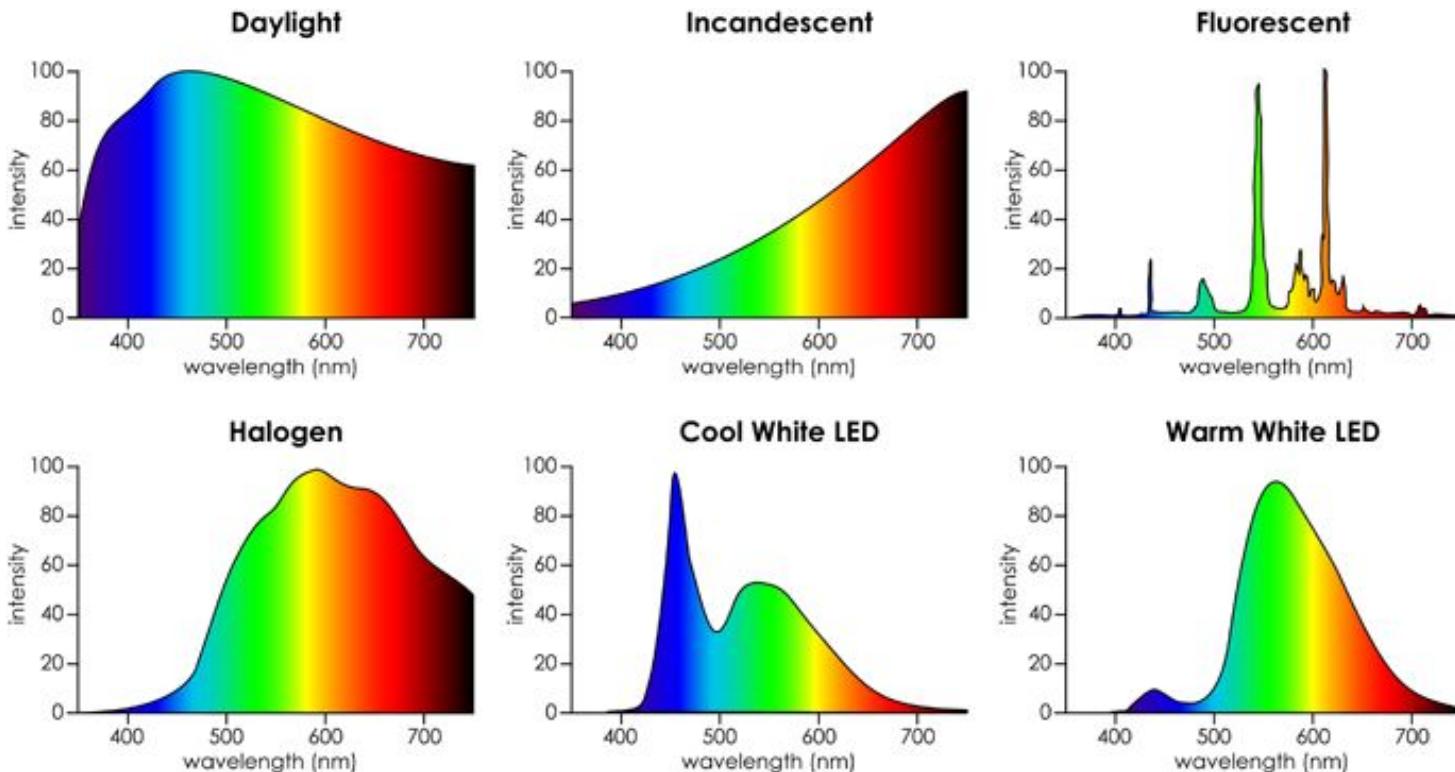
Su frecuencia es determinada por su temperatura.



Luz para caracterizar objetos

¿Cuánta y cuál luz emiten?

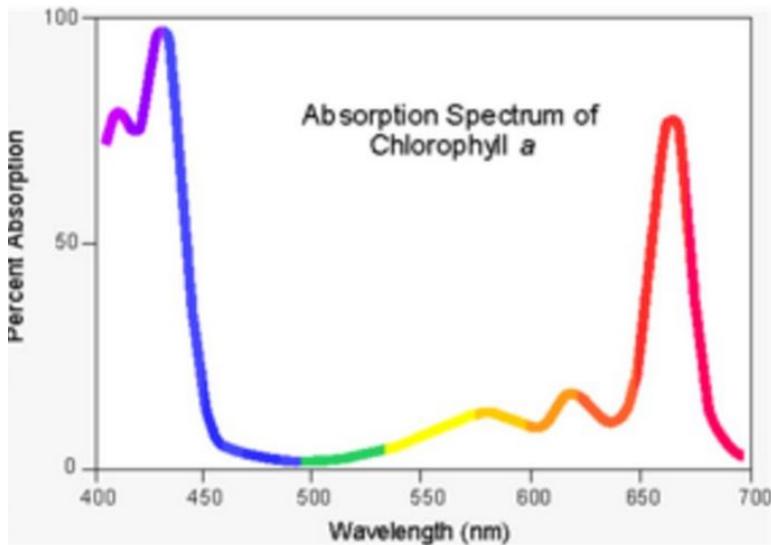
(esto es un modelo aditivo)



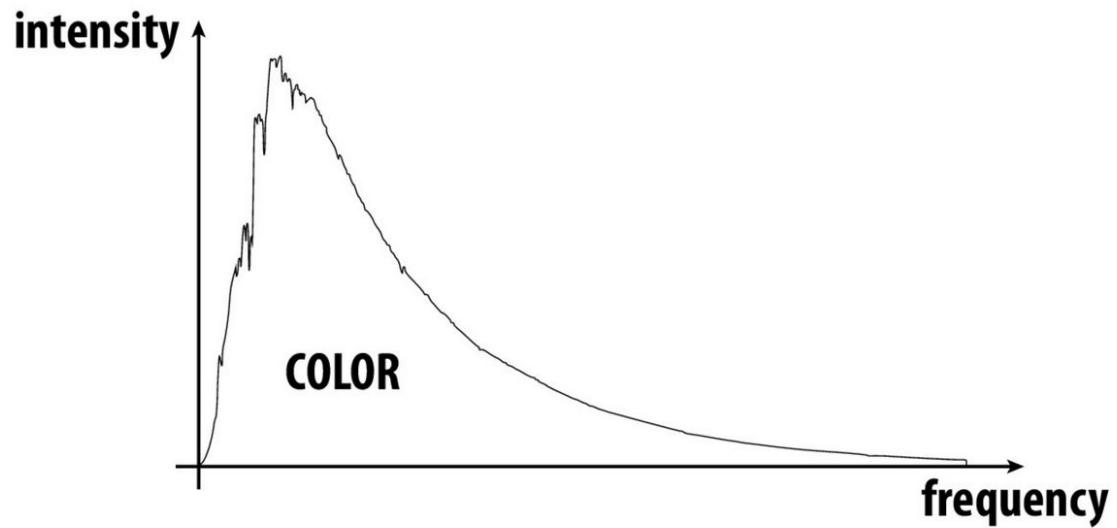
Luz para caracterizar objetos

Cuánta luz **absorben**

(esto es un modelo sustractivo)

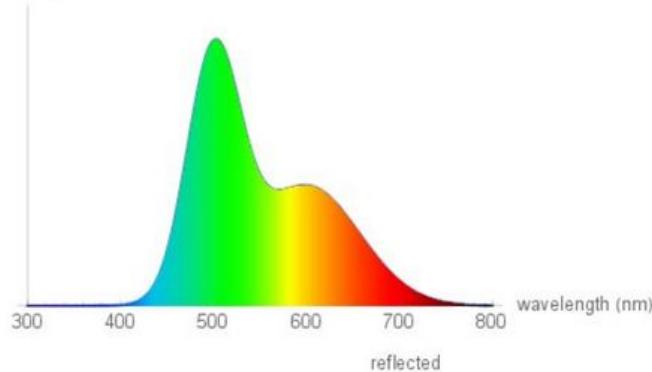


Entonces, la descripción fundamental del color es **la intensidad o la absorción de la luz como una función de la frecuencia de la onda.**



intensity

$f(v)$



% reflected

$g(v)$

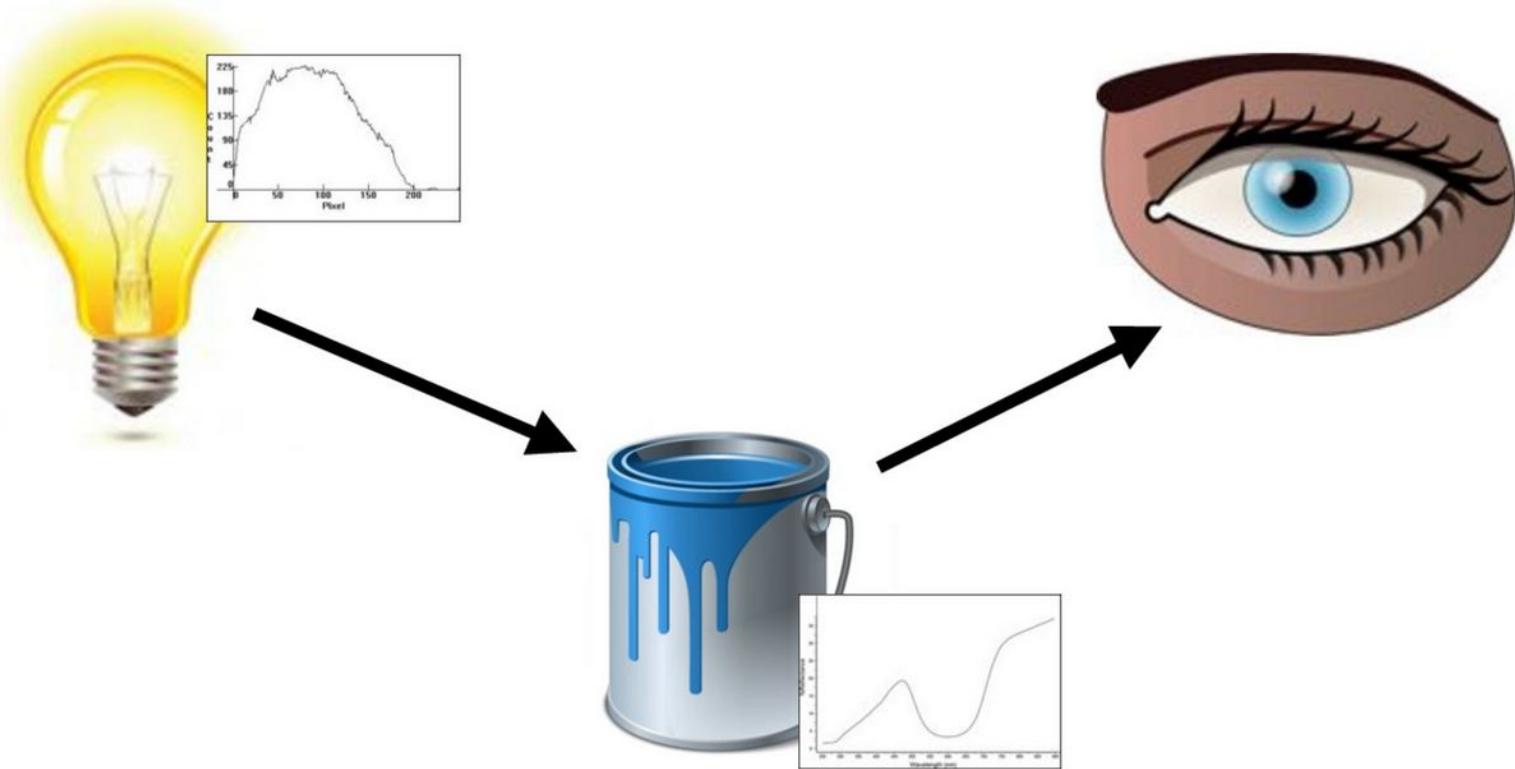


reflected

$f(v)g(v)$

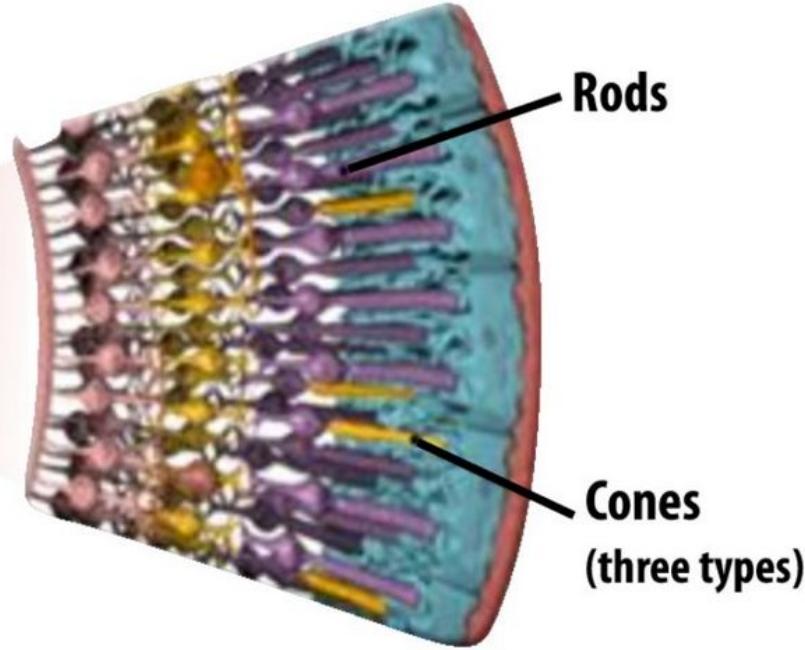
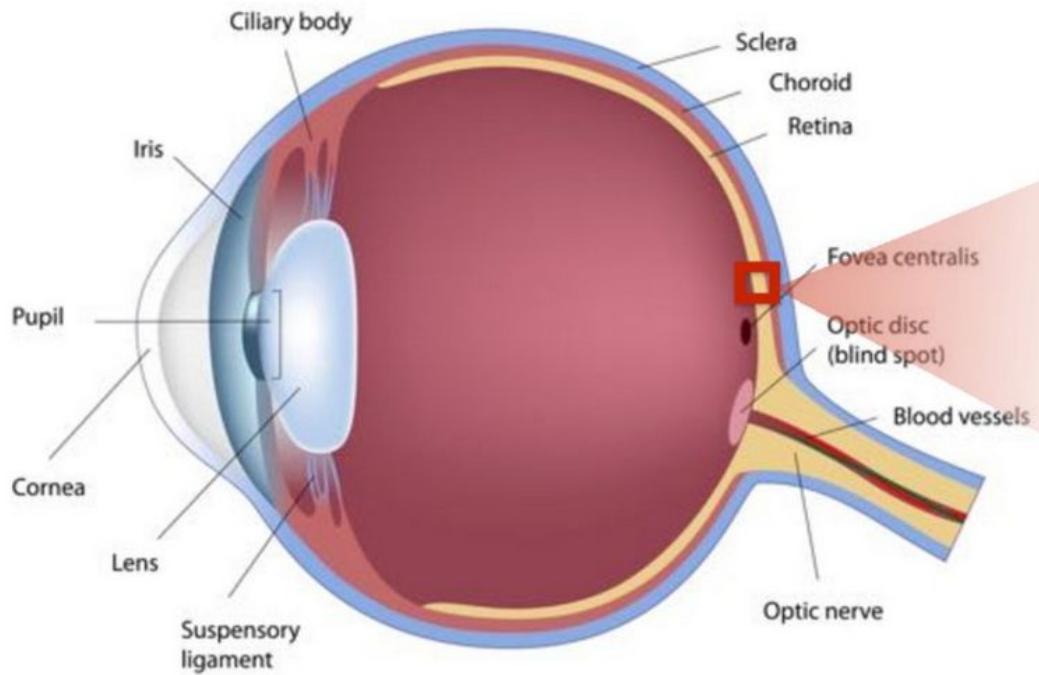


Por tanto, el color de un objeto se podría definir de manera ingenua como una función que combina ambas distribuciones.



¿Cuál es el color que observamos en la pintura?

¿Cómo le decimos al personal de la ferretería cuál es el color perfecto para nuestro hogar?

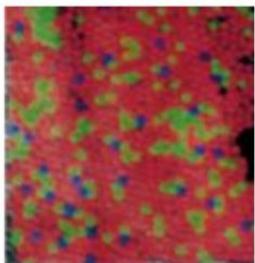


Nuestros ojos tienen aprox. 120 millones de **bastones** y entre 6 y 7 millones de **conos** (de tres tipos en total). Esos elementos fotorreceptores son los que procesan la luz que cruza la retina.

$$S = \int_{\lambda} \Phi(\lambda) S(\lambda) d\lambda$$

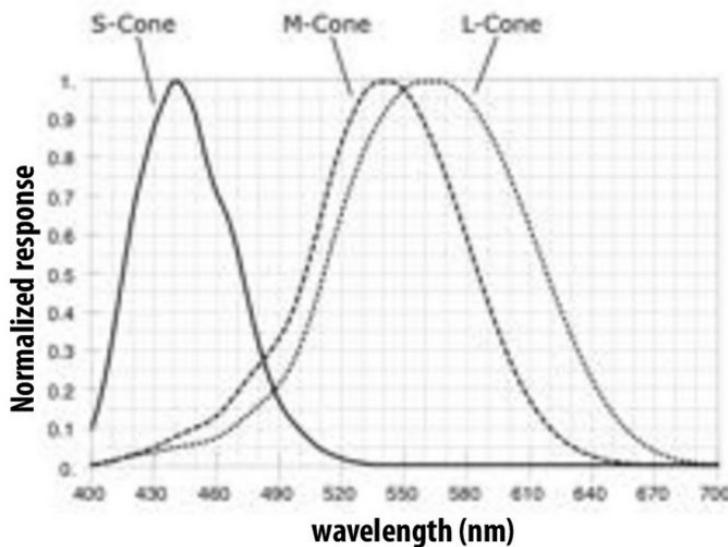
$$M = \int_{\lambda} \Phi(\lambda) M(\lambda) d\lambda$$

$$L = \int_{\lambda} \Phi(\lambda) L(\lambda) d\lambda$$



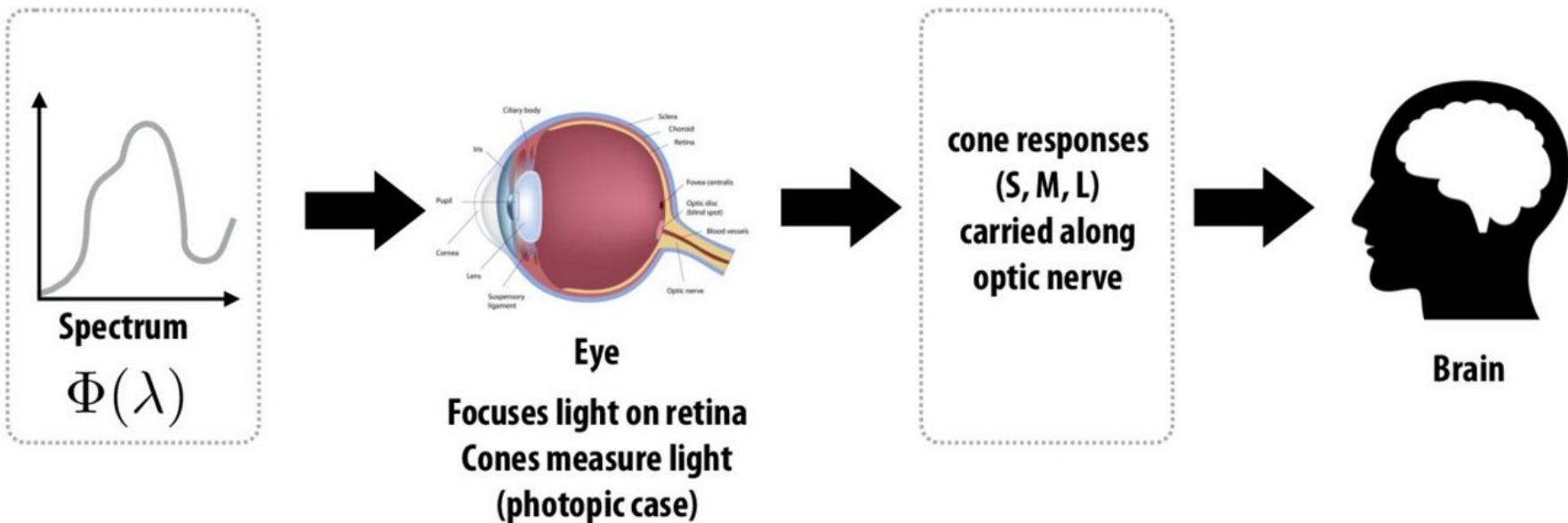
Uneven distribution of cone types in eye
~64% of cones are L cones, ~32% M cones

Response functions for S, M, and L cones



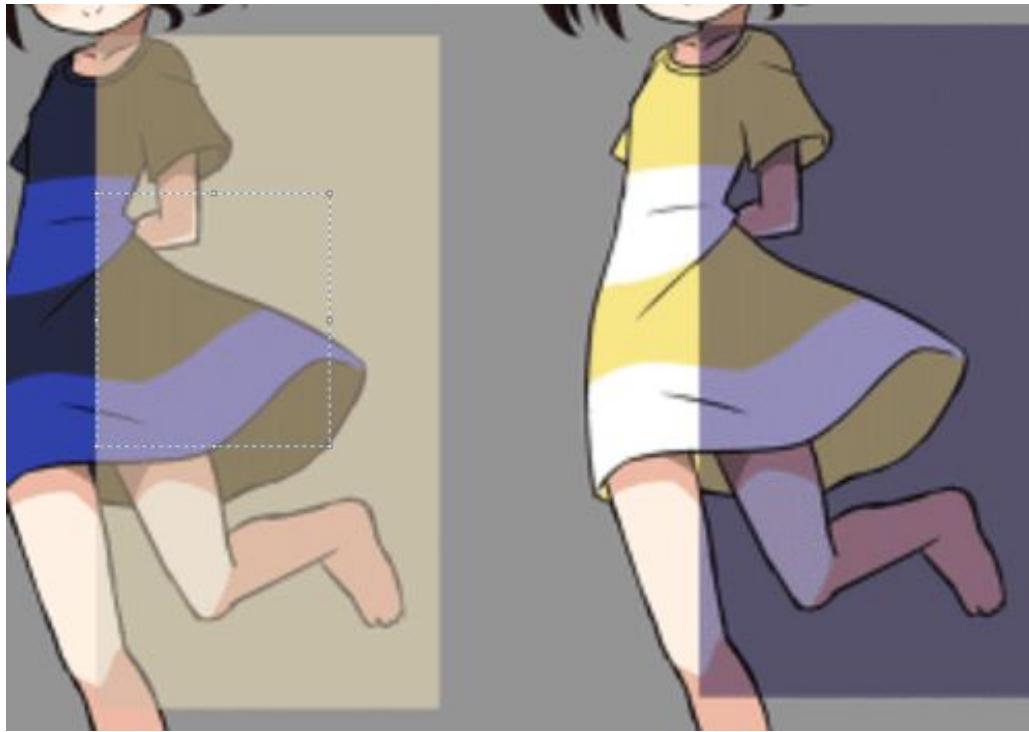
Cada tipo de cono reacciona ante distintos rangos de longitud de onda.

Esa es la física y la biología. ¿Qué pasa con la percepción?





¿Cuál es el color del vestido?

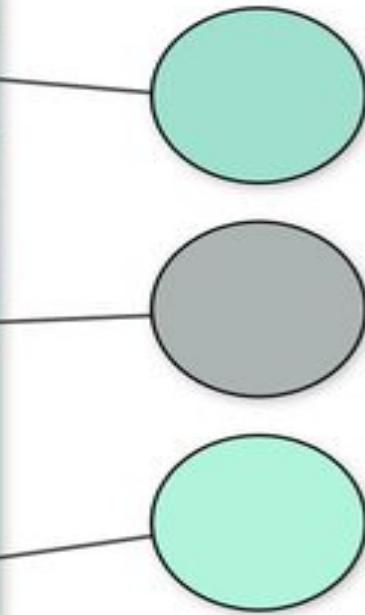
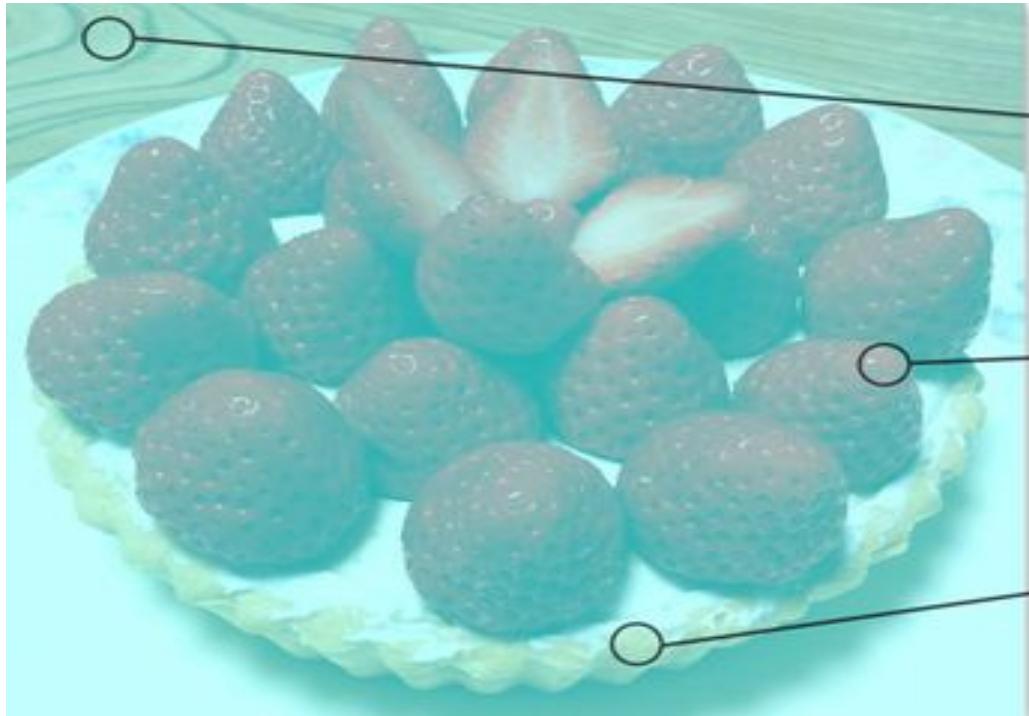


No es tan sencillo como aplicar el modelo, porque nuestra percepción también importa.

Hay que tener cuidado con los contextos y con la manera de percibir de nuestra mente.

¿De qué color son las frutillas?





En el curso trabajaremos con modelos de expresión de color.

La percepción la veremos al final en la unidad de visualización científica.

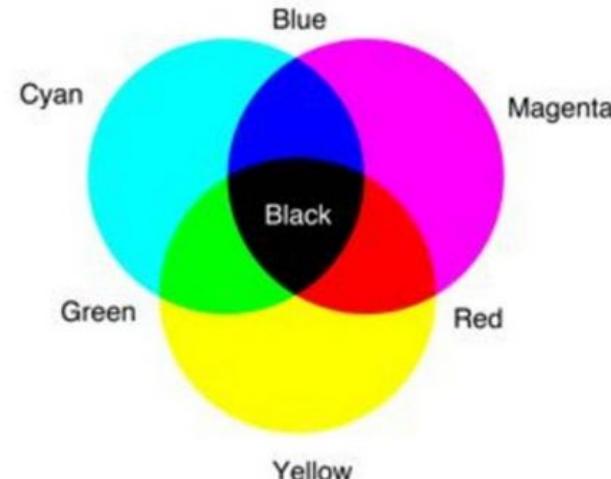
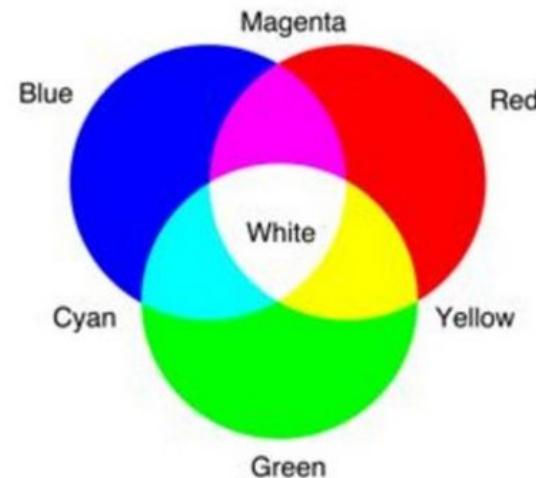
Modelos de color

RGB es un sistema aditivo (se añaden colores al negro), usado en pantallas.

CMY[K] es un sistema sustractivo (usado en impresoras).

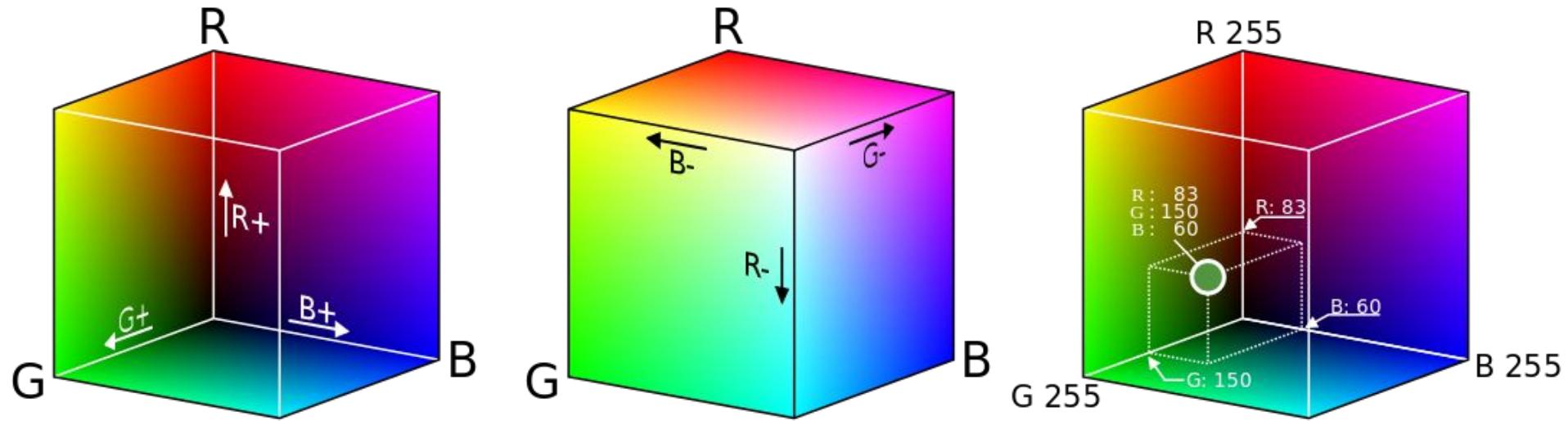
HSV es un espacio perceptual usado en arte, psicología y reconocimiento.

Implementación: La biblioteca **spectra** facilita operaciones y conversiones entre espacios: <https://github.com/jvine/spectra>



Cubo de color RGB

Cada valor (r , g , b) se expresa entre 0 y 255. Gris son valores en la diagonal. Colores “puros” en las esquinas.

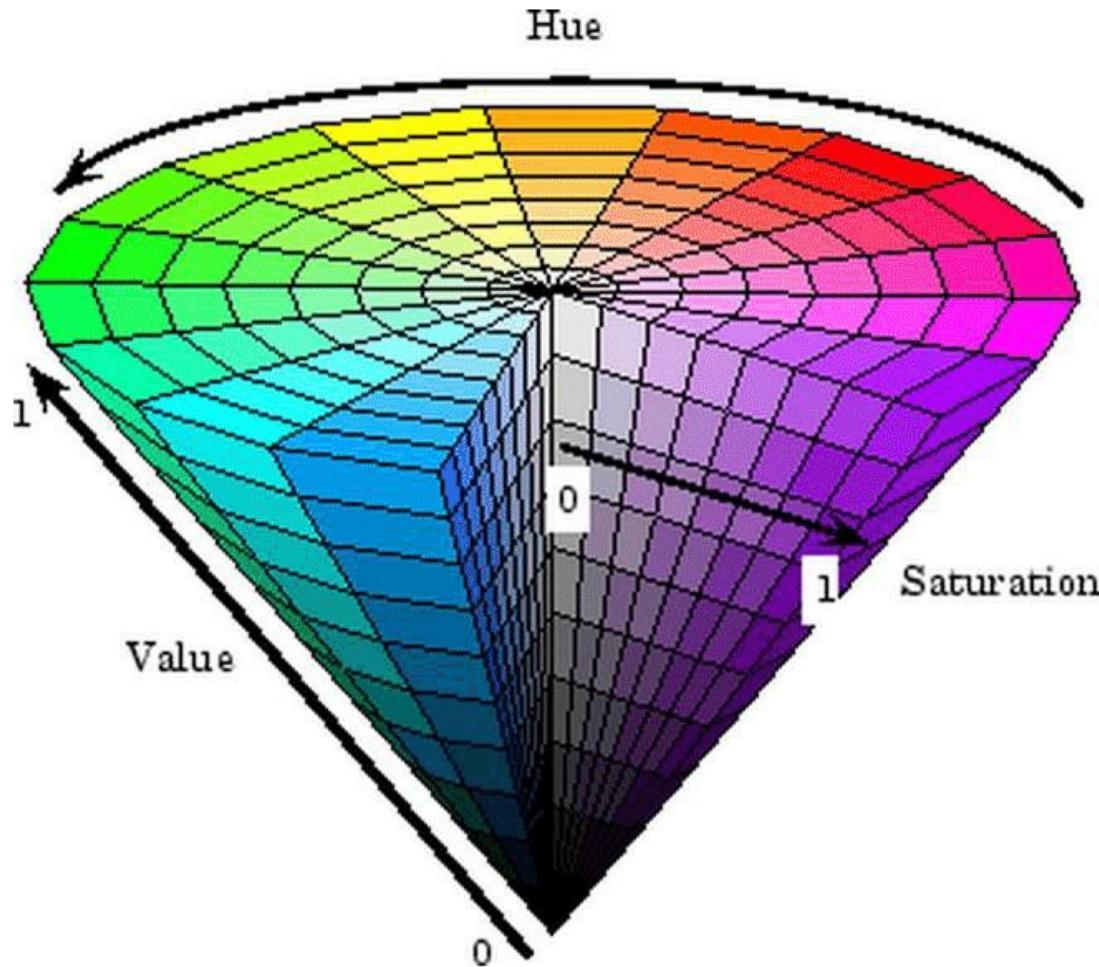


Hexágono de color HSV

Color codificado relativo a la diagonal del cubo de color. Hue es codificado como un ángulo, saturación es la distancia desde la diagonal y la intensidad es la altura.

HSV es una transformación de RGB. Por tanto, no es realmente un modelo que exprese colores distintos.

Sin embargo, es más sencillo para elegir un color. Es más **intuitivo**.



Modelo CMYK

Cyan, magenta, yellow, black

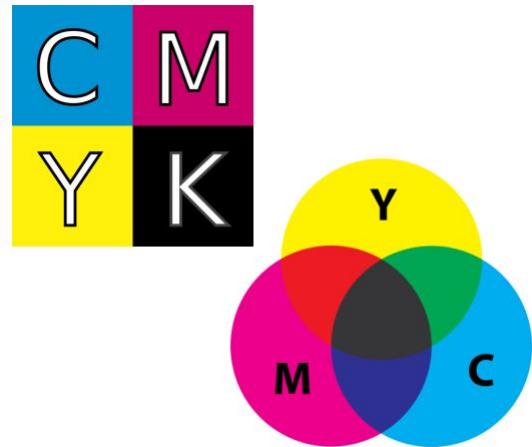
Modelo sustractivo de color

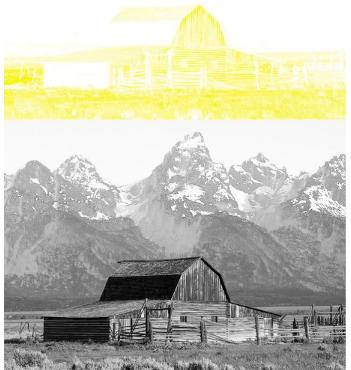
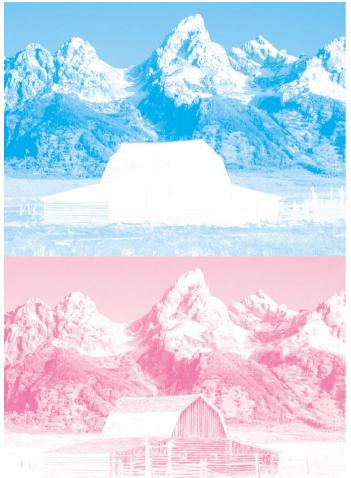
- Sustrae colores de la luz

¿Cómo creen que lo hace?

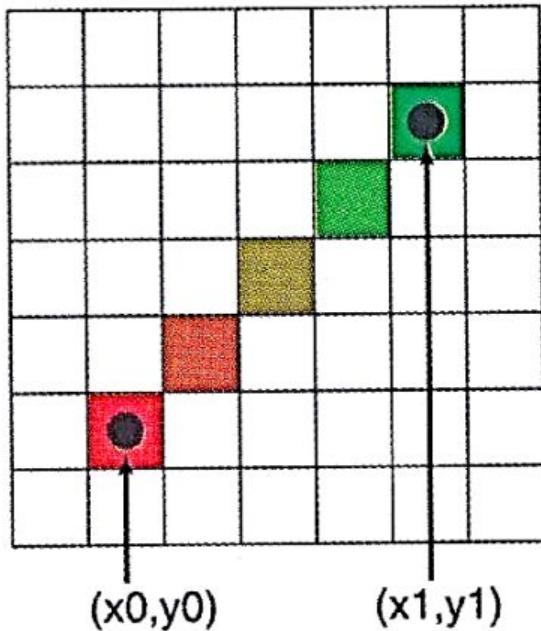
Utilizado en impresión

- Utiliza tinta negra en vez de los otros tres colores
- **¿Por qué creen que se necesita tinta negra extra?**





Interpolación de colores en RGB



$(r,g,b) = (0.00, 1.00, 0.00), t = 1.00$

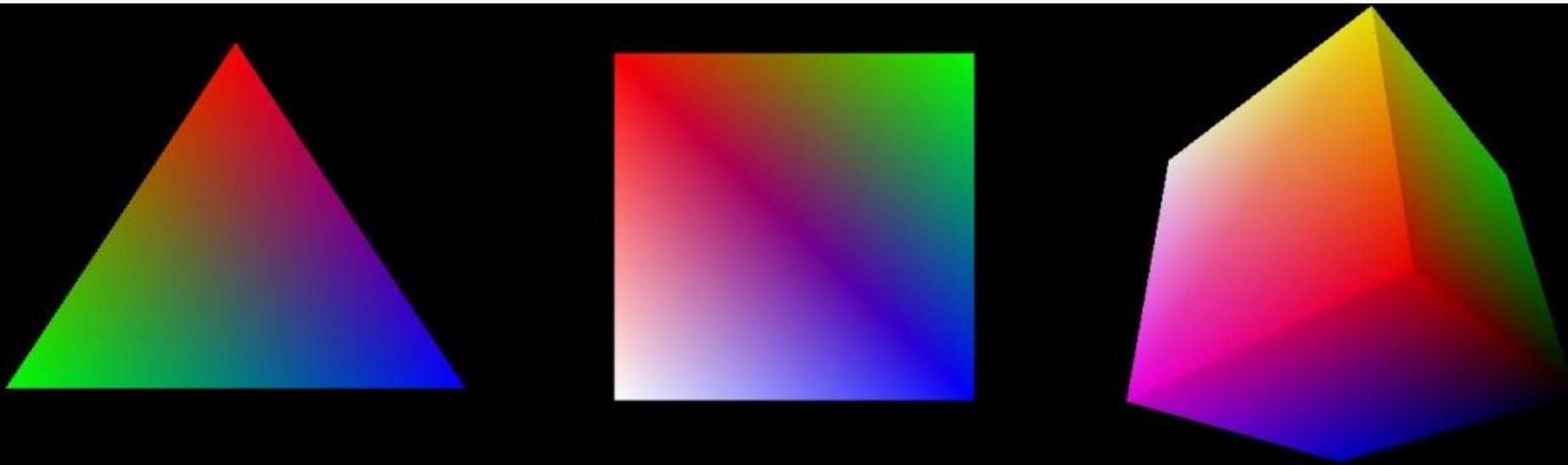
$(r,g,b) = (0.25, 0.75, 0.00), t \approx 0.75$

$(r,g,b) = (0.50, 0.50, 0.00), t = 0.50$

$(r,g,b) = (0.75, 0.25, 0.00), t = 0.25$

$(r,g,b) = (1.00, 0.00, 0.00), t = 0.00$

Interpolación de colores

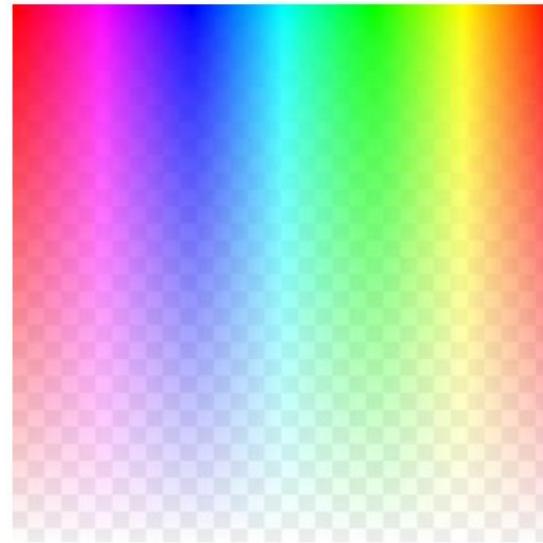


El canal alfa para transparencia

Cada píxel necesita

- Rojo: 8 bits
 - Verde: 8 bits
 - Azul: 8 bits
 - +Alfa: 8 bits
 - Total: 32 bits

Si un color tiene transparencia, se necesita conocer el color del fondo.



$$c = \alpha c_f + (1 - \alpha) c_b$$

foreground background

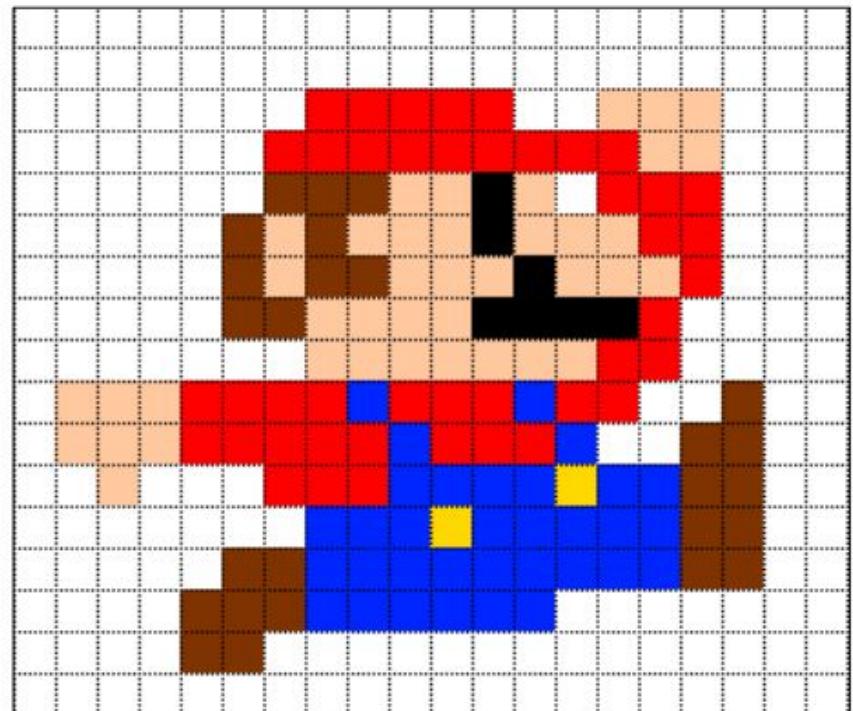
¿Cómo se representa una imagen en un computador?

Una matriz 2D donde cada celda tiene asignado un color.

Cada celda se conoce como *pixel*.

Este modelo de visualización se conoce como *raster*.

La imagen que se muestra en pantalla se almacena en un *frame buffer*.



Créditos: Nintendo

Imágenes Raster

Modelo simple

- Un píxel por celda

Resolución fija

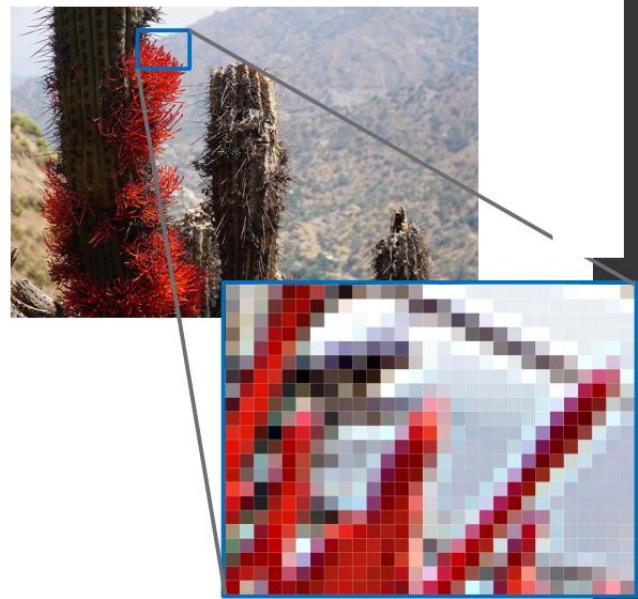
- Se pierde nitidez al hacer zoom

Costoso de almacenar

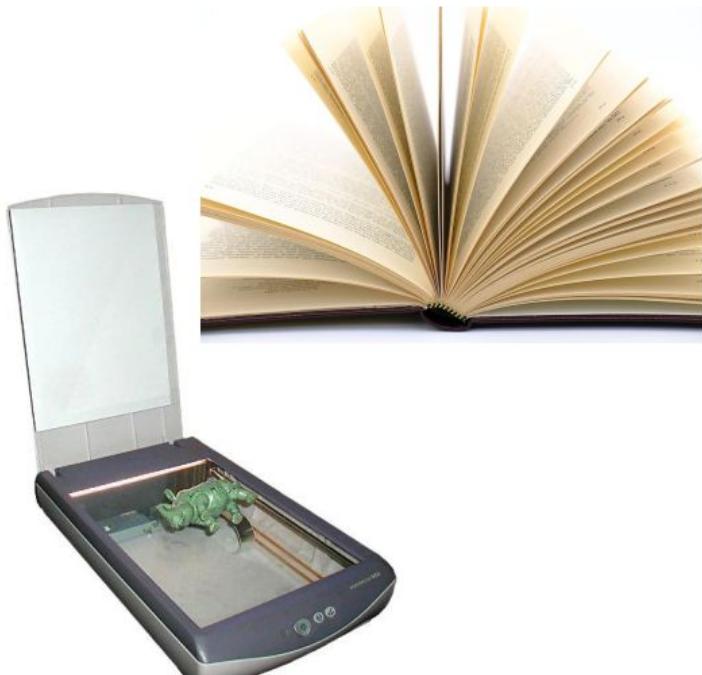
- Distintas estrategias de compresión

Formatos

- JPG (compresión con pérdida), PNG (compresión sin pérdida), BMP (sin compresión)



Dispositivos Raster



Imágenes vectoriales

Modelos **paramétricos** por cada figura representada

- Líneas, cuadrados, círculos, curvas, etc.

Resolución infinita:

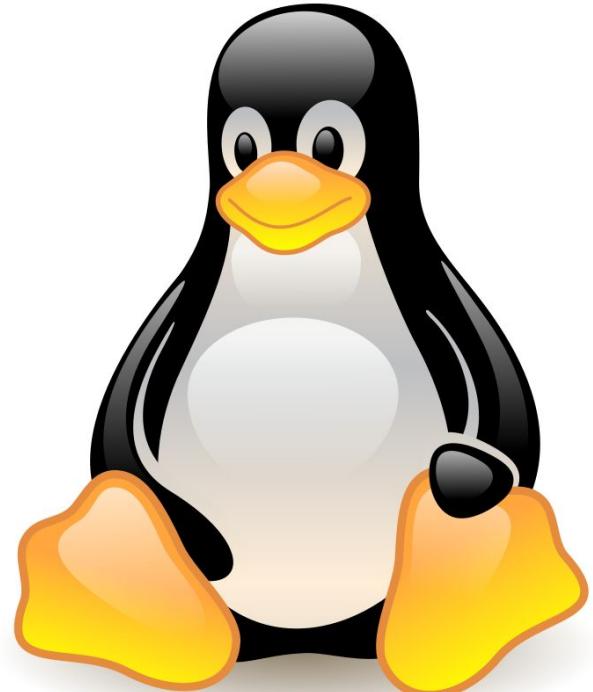
- No hay pérdida de nitidez al hacer zoom, porque se evalúan las funciones que generan el gráfico

Tamaño depende de la **complejidad del contenido**

Háganle **zoom** a Tux:

<https://upload.wikimedia.org/wikipedia/commons/b/b0/NewTux.svg>

Formatos: **SVG (Scalable Vector Graphics)**, EPS, PDF



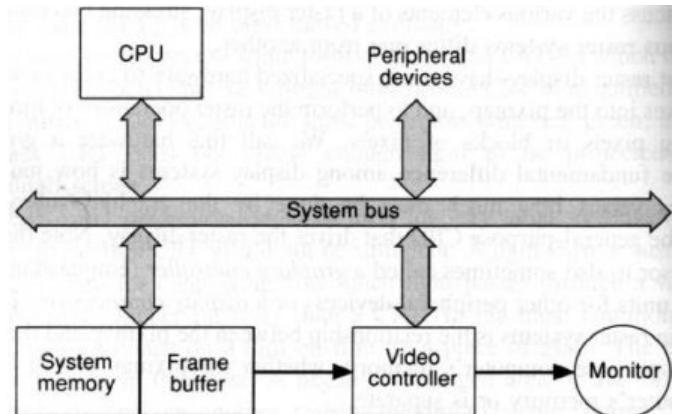
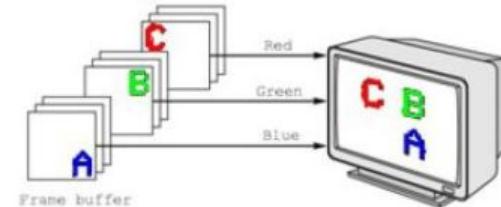
Arquitectura

Framebuffer

- Área de memoria para almacenar la imagen.
- En ocasiones hay dos: el del frame actual, y el del frame siguiente. El actual se grafica, el siguiente se modifica.

Video Controller

- Accesa el framebuffer para refrescar la pantalla.
- Obtiene valores de los pixels durante un ciclo de refresco (60 cuadros por segundo es lo típico).
- Contiene la tabla de colores a utilizar
 - Esto cuando la paleta de colores es limitada.
 - Hoy puedes calcular el color en la GPU utilizando un **fragment shader**.



¿Cómo especificar un color?

Almacenar directamente el color en el framebuffer.

Ejemplo: Framebuffer de 3 bits

- Primer bit para el rojo
- Segundo bit para el verde
- Tercer bit para el azul
- Podemos representar 8 colores

Color Code	Stored Color Values in Frame Buffer			Displayed Color
	RED	GREEN	BLUE	
0	0	0	0	Black
1	0	0	1	Blue
2	0	1	0	Green
3	0	1	1	Cyan
4	1	0	0	Red
5	1	0	1	Magenta
6	1	1	0	Yellow
7	1	1	1	White

¿Cómo especificar un color? Esquema directo

Para representar más colores, se **requieren más bits**

Simple, pero **costoso en memoria**

¿Cuánta memoria se necesita para un frame buffer de **1920x1080** pixels (Full HD) donde cada color utiliza 8 bits?

¿Y si fueran 32 bits?

¿Y en una pantalla 4K?

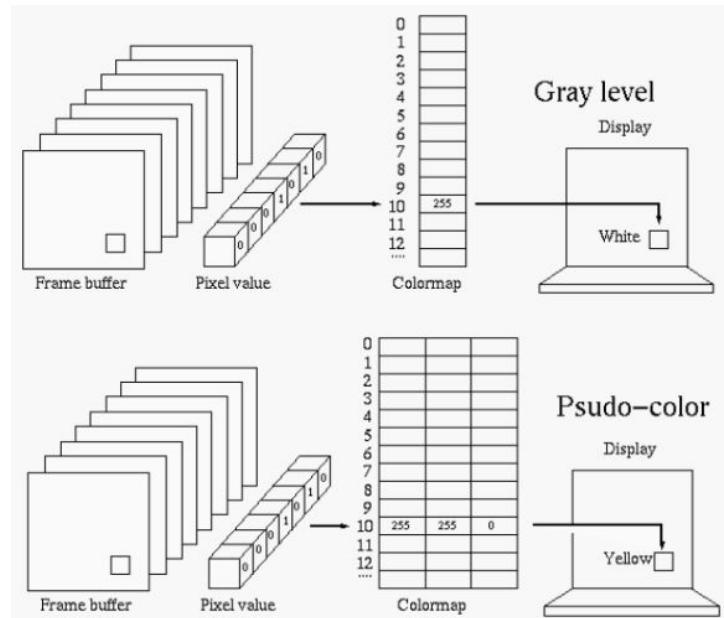
¿Y con dos monitores?

¿Cómo especificar un color? Esquema indirecto

Almacenar el color a utilizar en una tabla separada,

En el framebuffer se almacena un índice de un color almacenado en la tabla.

Así funcionaban las consolas hasta la generación del **Super Nintendo**. Si bien usaba 15 bits para representar un color, usaba una tabla de 256 colores (que tú podías manipular).



Esquema Indirecto

Es posible cambiar el color asociado a un índice:

- Misma imagen puede ser coloreada distinto



True Color



False Color



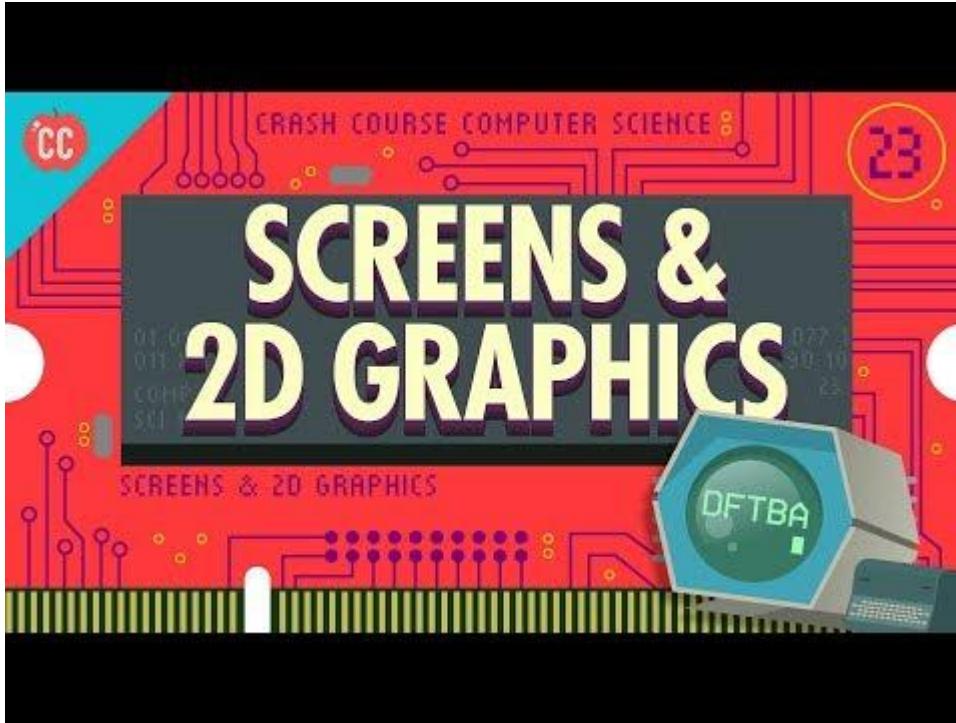


Arte y magia con paletas de colores: Mark Ferrari

- Imágenes digitales animadas [[link](#)]
- Presentación en Game Developer Conference (GDC) [[link](#)]



Crash Course: Screen & 2D Graphics



A brief history of graphics



¿Preguntas?

Créditos: basado en material de Iván Sipirán y Nancy Pollard