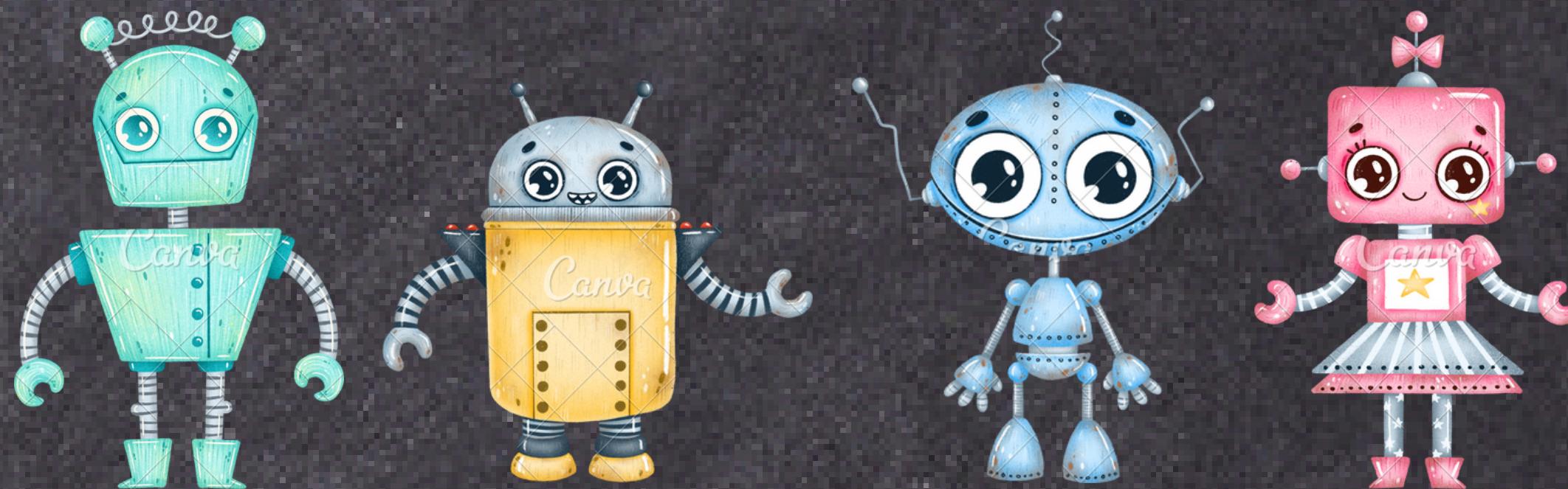


FILTROS EN IMÁGENES CON TRANSFORMADA DE FOURIER 2D (PASA-BAJAS VS. PASA-ALTAS)

UNA IMPLEMENTACIÓN INTERACTIVA EN PYTHON

ALUMNOS:

CASTILLO RIVERA DIEGO
REYES SANATAMARÍA TANIA JANNET



MATERIA: MATEMÁTICAS AVANZADAS
PROFESOR: DR. DAVID CORREA COYAC
INSTITUCIÓN: IPN - ESCOM.

INTRODUCCIÓN Y OBJETIVO

¿QUÉ HICIMOS?

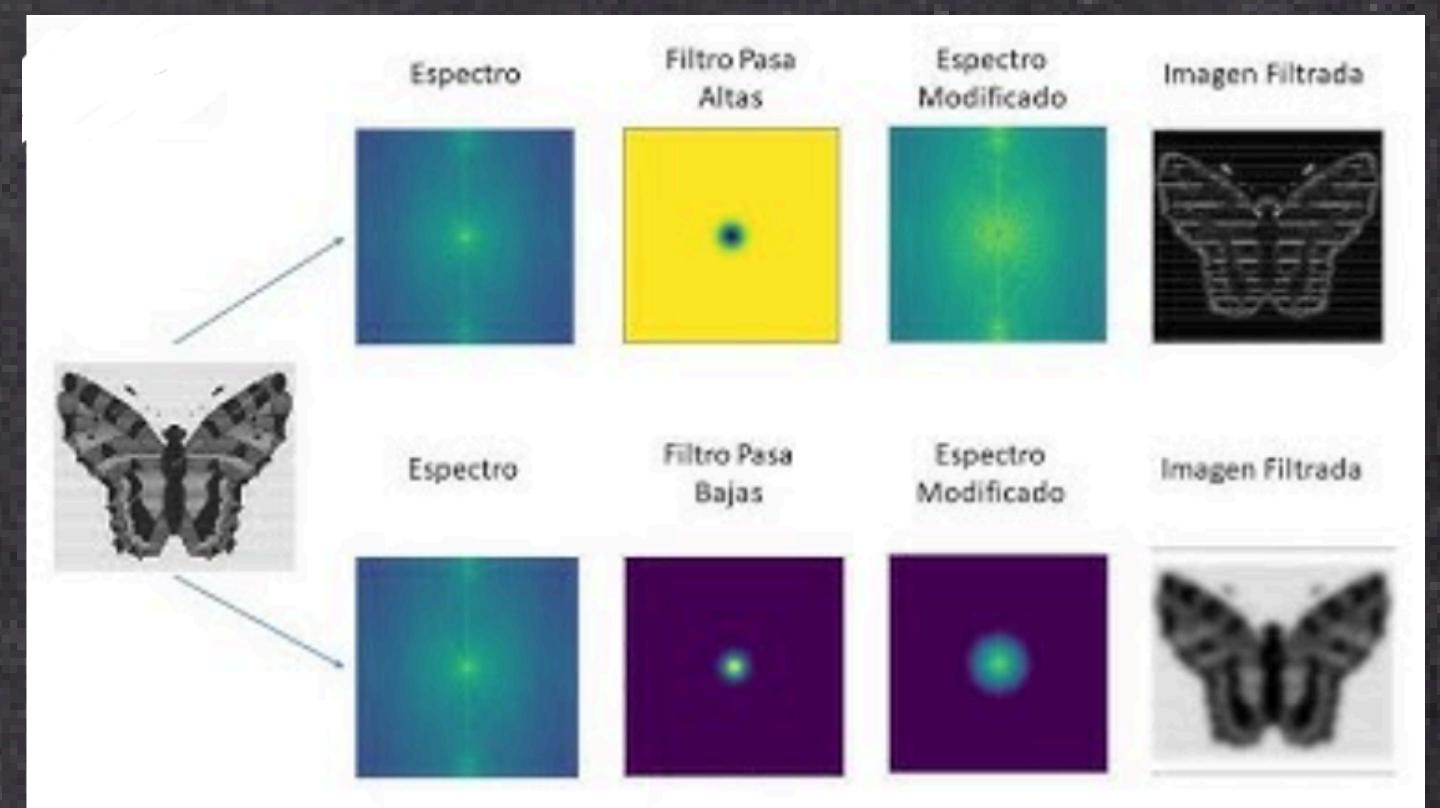
Desarrollamos una aplicación en Python para descomponer imágenes en sus frecuencias y filtrarlas.

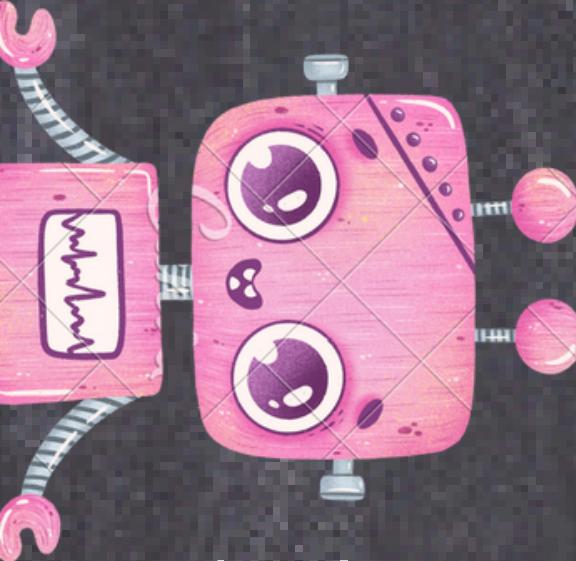
¿POR QUÉ ES IMPORTANTE?

Entender la "magia matemática" que ocurre detrás de las librerías de procesamiento de imágenes.

OBJETIVO:

Comparar visual y matemáticamente cómo un Filtro Pasa-Bajas (suavizado) se diferencia de un Filtro Pasa-Altas (bordes).





TRANSFORMADA DISCRETA DE FOURIER EN 2D (DFT)

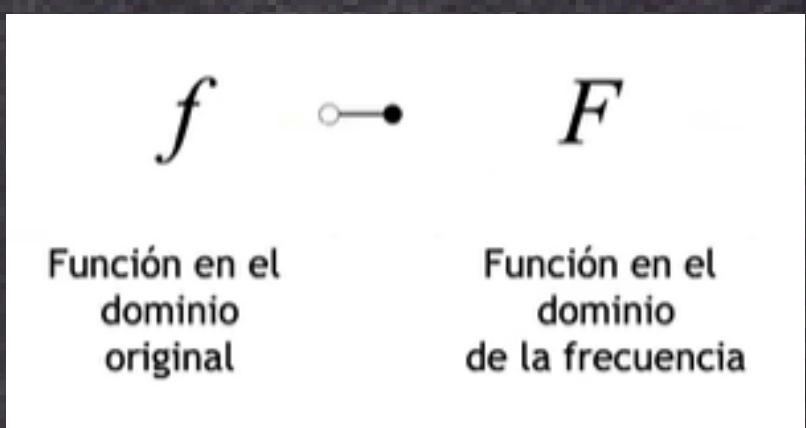
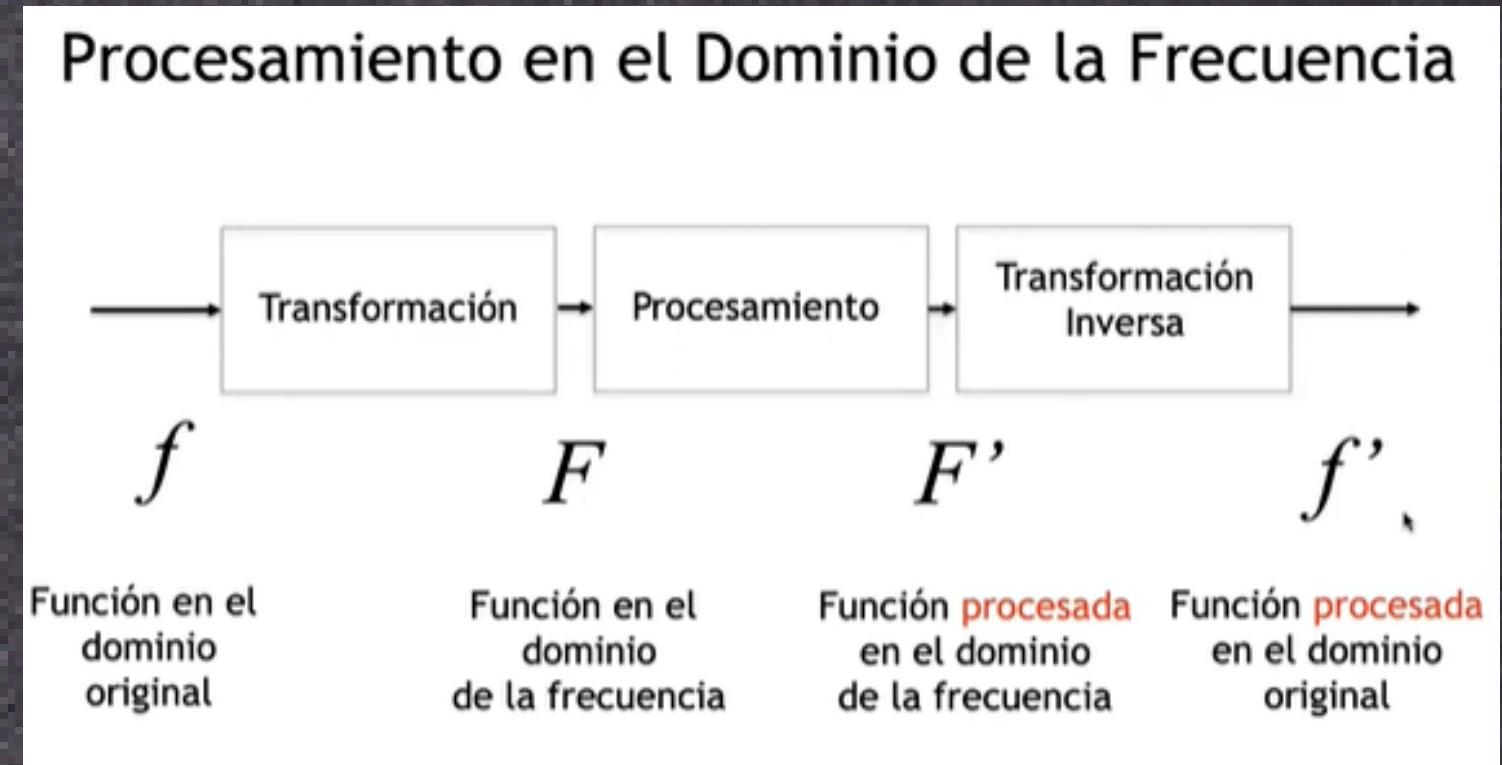
La imagen se representa como una función discreta $f(x,y)$ de tamaño $M \times N$. La DFT convierte esta información espacial en el dominio de la frecuencia $F(u,v)$.

FÓRMULA:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

¿QUÉ HACE ESTA FÓRMULA?

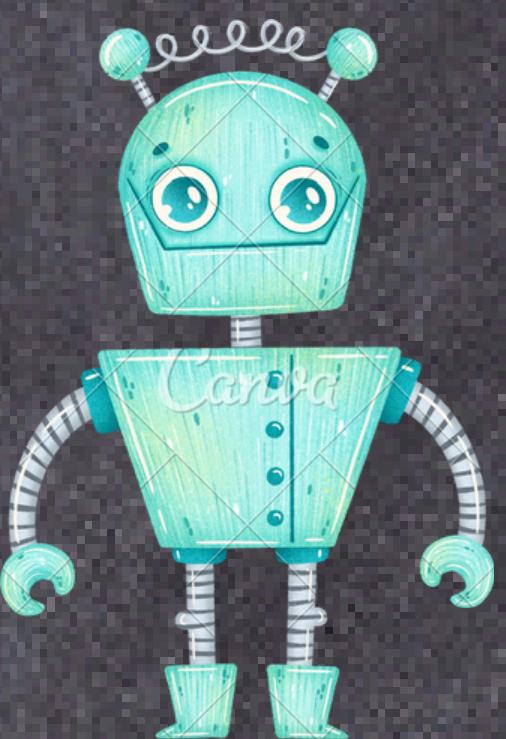
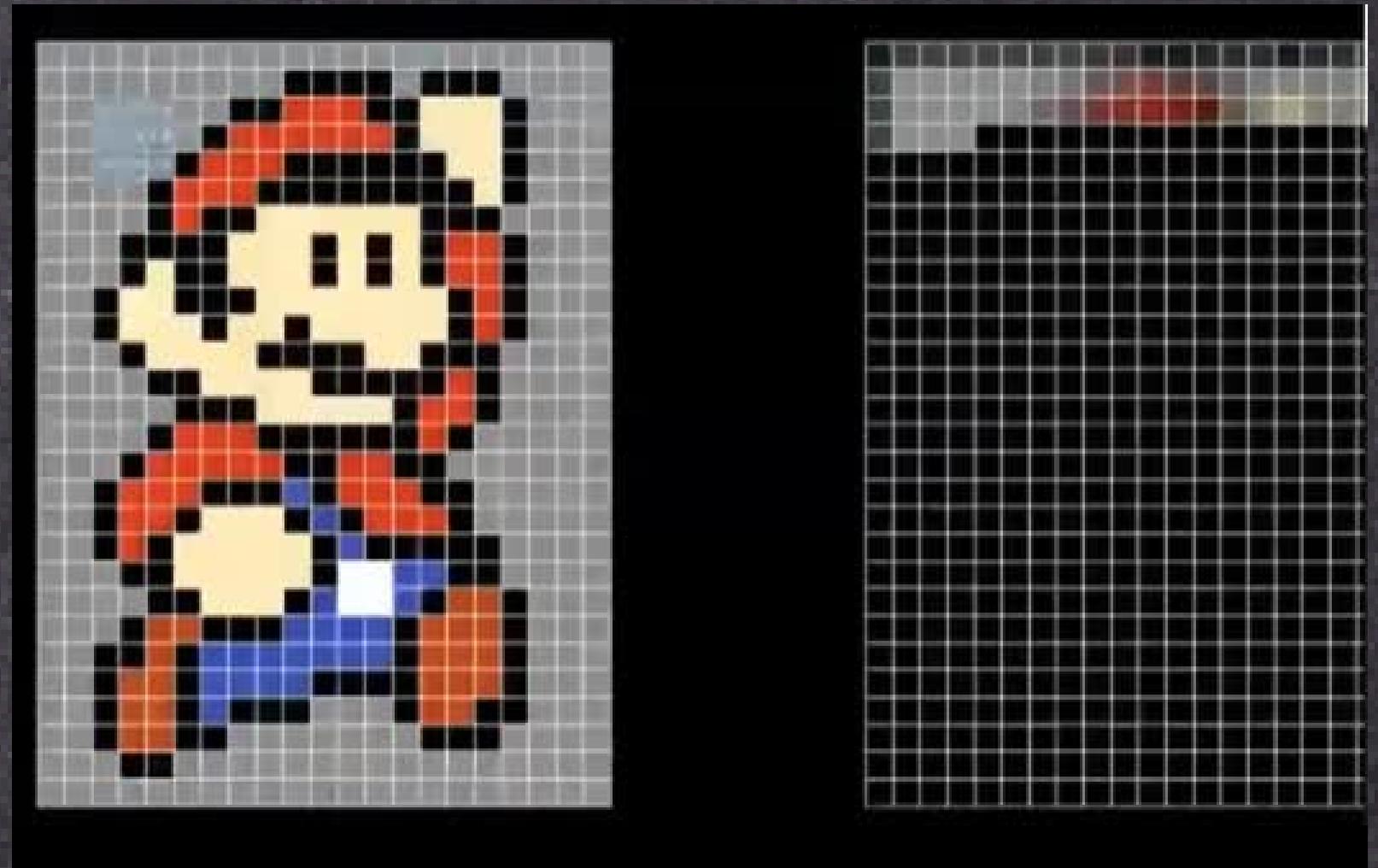
Convierte información de espacio (píxeles) a frecuencia (senos y cosenos)



CONVOLUCIÓN

¿QUÉ ES?

Operación matemática donde un pequeño kernel (matriz de números) se desliza sobre una imagen, realizando multiplicaciones y sumas con los píxeles vecinos para producir una imagen de salida, se usa para tareas como desenfocar, realzar bordes, reducir ruido o detectar objetos mediante la extracción de características, transformando la imagen original.



CONVOLUCIÓN

¿CÓMO LO IMPLEMENTAMOS EN NUESTRO PROYECTO?

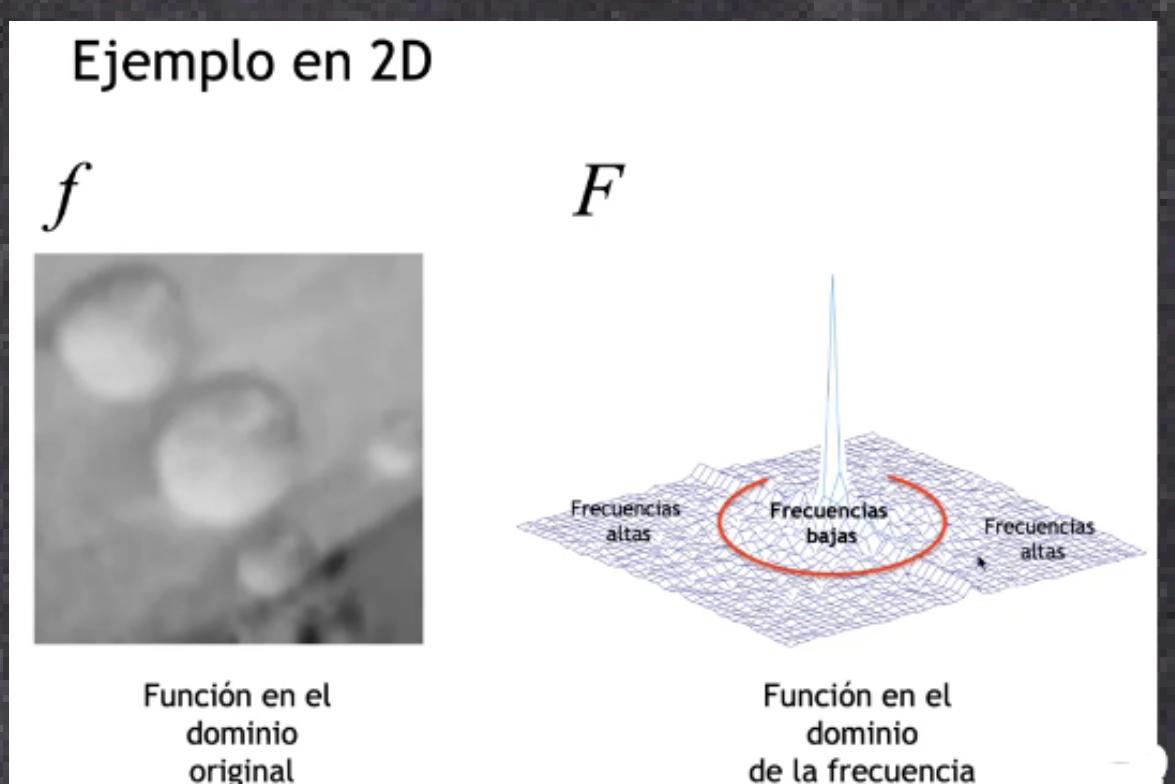


El Problema: Filtrar en el espacio (hacer convolución píxel por píxel) es lento y costoso computacionalmente.

La Solución: El Teorema de la Convolución.

$$f(x, y) * h(x, y) \iff F(u, v) \cdot H(u, v)$$

En lugar de convolucionar, simplemente multiplicamos la imagen transformada por una máscara (el filtro) y es mucho más rápido.



FILTROS (MÁSCARAS)

Preparación (Shift): Primero centramos las frecuencias bajas en el medio de la imagen usando la propiedad de traslación: $(-1)^{x+y}$.

Filtro Ideal: Usamos un círculo definido por una distancia D_0 (Radio de corte).

Pasa-Bajas: Deja pasar lo que está ADENTRO del círculo (frecuencias bajas).

Resultado: Imagen borrosa/suave.

Pasa-Altas: Deja pasar lo que está AFUERA del círculo (frecuencias altas).

Resultado: Solo bordes y detalles finos.

Filtro pasa-bajas

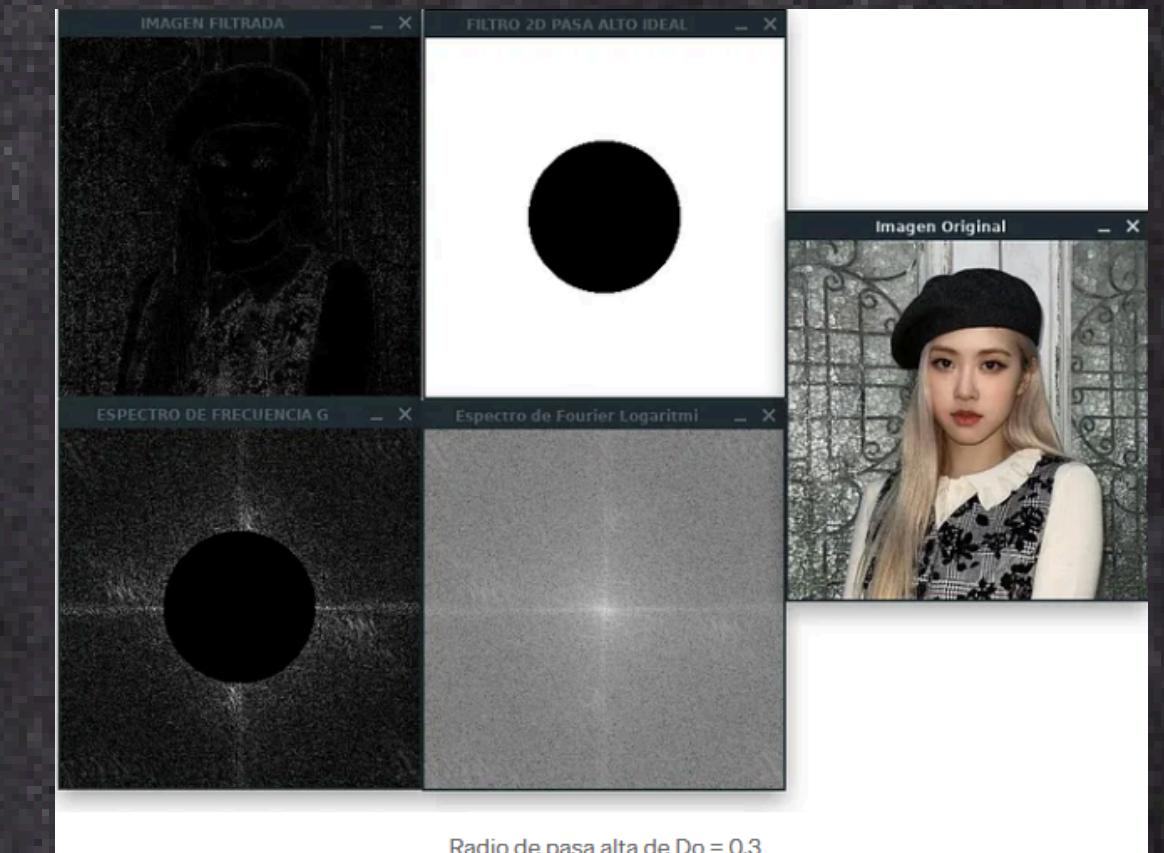
Imagen de entrada -



Salida:



Filtro pasa-altas



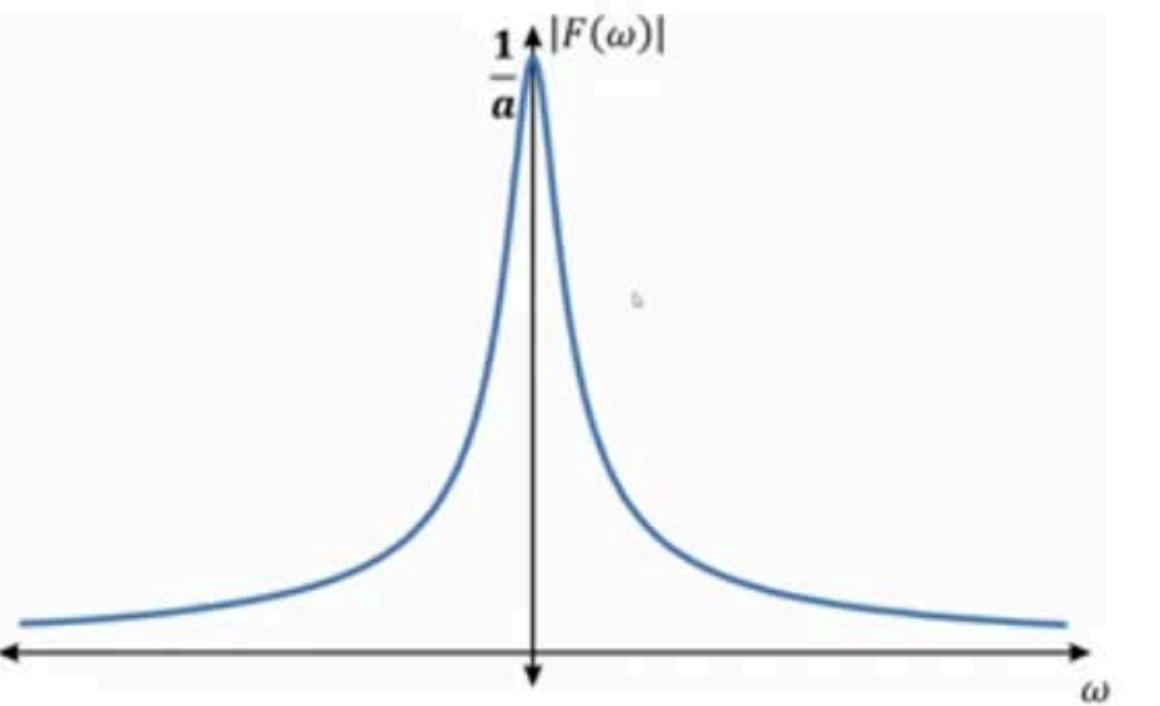
Radio de pasa alta de $D_0 = 0.3$

TRANSFORMADA DE FOURIER ESPECTROS DE MAGNITUD Y FASE

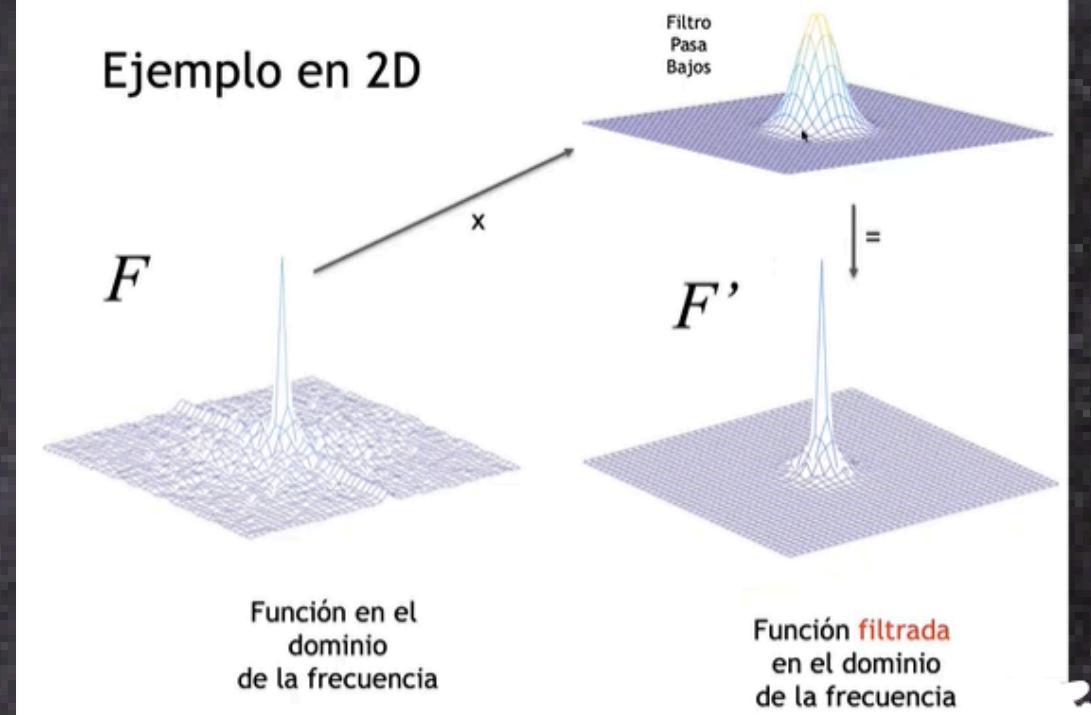
$u(t)e^{-at}$

Espectro Magnitud

$$|F(\omega)| = \frac{1}{a} \cdot \frac{1}{\sqrt{\left(1 + \frac{\omega^2}{a^2}\right)}} \quad a > 0$$

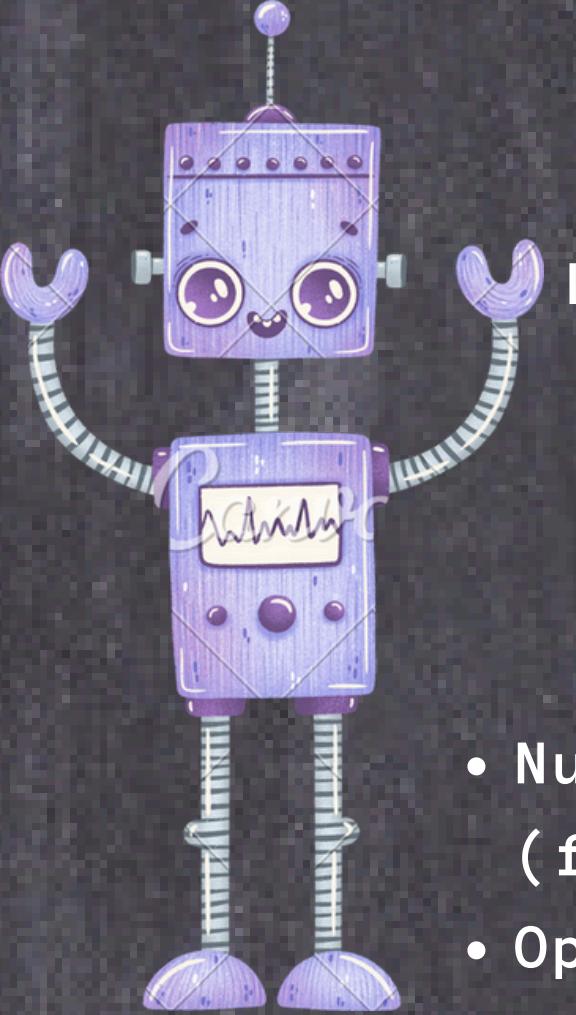


Ejemplo en 2D



Matfís Academy





ARQUITECTURA DE LA IMPLEMENTACIÓN

LENGUAJE: Python (por su poder de procesamiento de operaciones matemáticas debido a sus librerías).

LIBRERÍAS CLAVE:

- NumPy: Para las matrices y la FFT (`fft2`, `fftshift`).
- OpenCV: Para leer la imagen y pasar a escala de grises.
- Matplotlib: Para graficar y crear los Sliders interactivos.

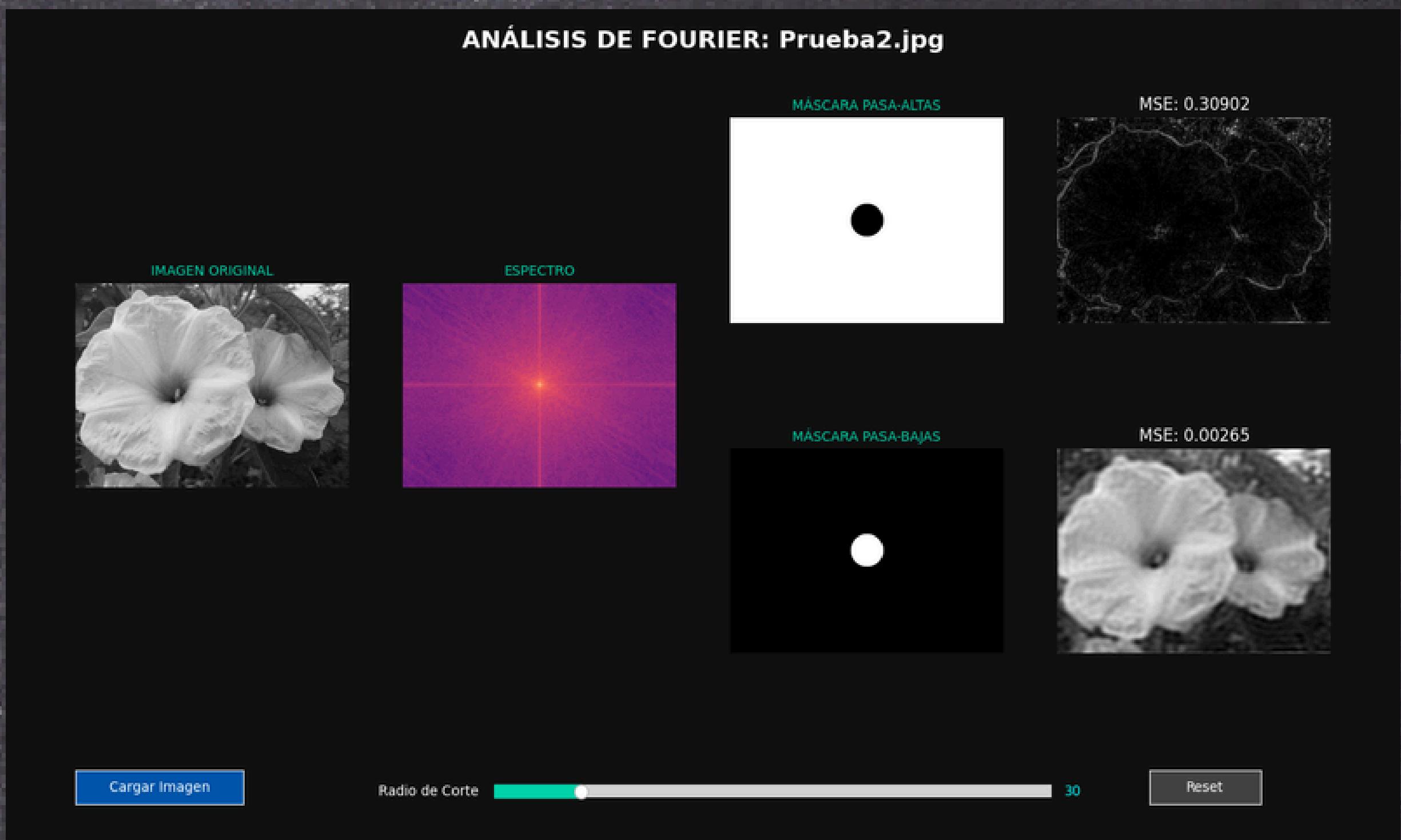
FLUJO DE TRABAJO:

- Cargar Imagen.
- Escala de Grises.
- Normalizar.
- Aplicar FFT.
- Centrar (Shift).
- Multiplicar por Máscara (Círculo).
- Inversa (IFFT).
- Imagen Filtrada.

PREPROCESAMIENTO IMPORTANTE

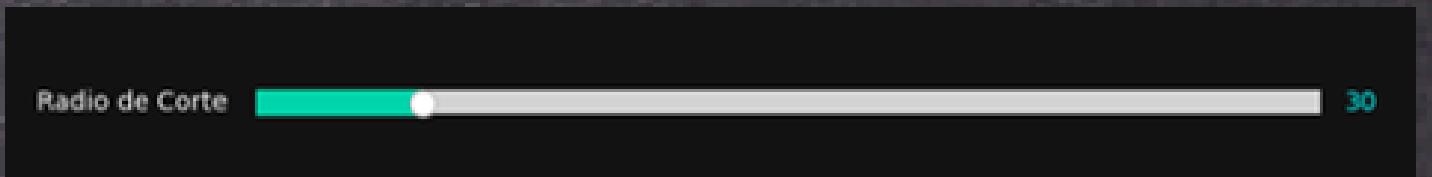
- ESCALA DE GRISES: Se eliminan los canales RGB para trabajar con una señal 2D pura de intensidad.
- NORMALIZACIÓN: Se divide entre 255.0 para tener valores de 0 a 1. Esto es vital para que las matemáticas de la FFT no se desborden y se visualicen bien.

RESULTADOS

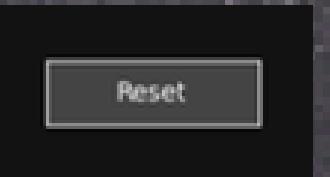


INTERACTIVIDAD:

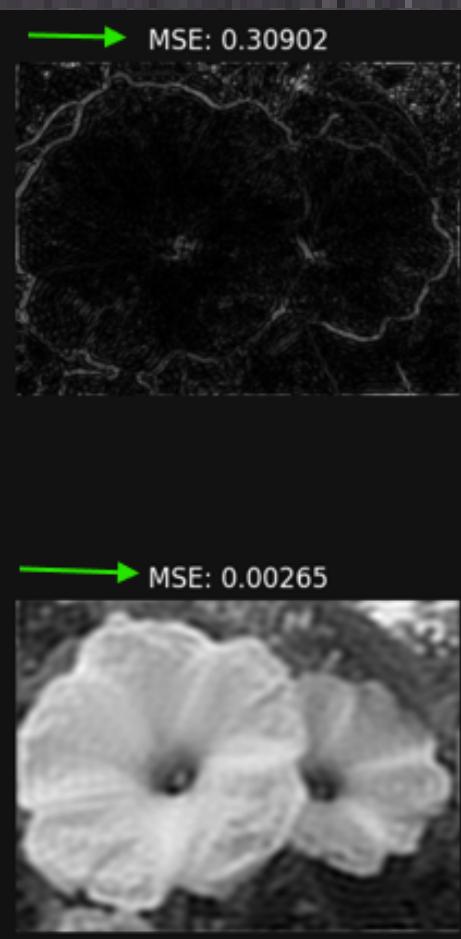
El uso de Sliders permite cambiar el radio D0 en tiempo real.



El botón: “Resetear” permite que el radio D0 regrese al valor predeterminado (30).

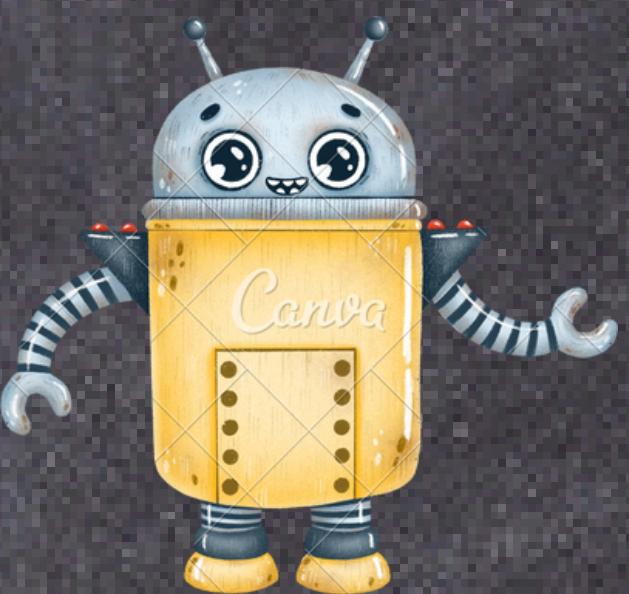


El botón: “Cargar archivo” permite que el usuario cargue una nueva imagen para que el programa la procese, sin necesidad de que vuelva a ejecutar el código.



Métrica MSE (Error Cuadrático Medio): Calculamos qué tanta información se perdió.

- Pasa-Bajas: MSE bajo (se parece más a la original).
- Pasa-Altas: MSE alto (se pierde casi toda la información de color/relleno).



CONCLUSIÓN

- La FFT nos permite manipular imágenes basándonos en su contenido de frecuencia y no solo en sus píxeles.
- Los Filtros Pasa-Bajas eliminan ruido pero pierden detalle.
- Los Filtros Pasa-Altas detectan bordes pero introducen ruido.
- El teorema de la convolución hace que este procesamiento sea eficiente.

