

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

PROCESAMIENTO DIGITAL DE IMÁGENES

PRÁCTICA 3

**OPERACIONES DE REALCE UTILIZANDO EL
HISTOGRAMA: TRANSFORMACIONES COMÚNES
Y EQUALIZACIÓN.
FILTRADO ESPACIAL**

23 de octubre de 2023

PROFESORA:

Dra. María Elena Martínez Pérez

AYUDANTE:

Miguel Angel Veloz Lucas

ALUMNA:

Janet Illescas Coria

Objetivo

- Realizar diversas modificaciones al histograma con transformaciones básicas como son: negativo, exponencial, logarítmica y gamma.
- Ecualización del histograma de una imagen.
- Realizar operaciones de suavizado y de reducción de ruido en imágenes utilizando filtros espaciales.
- Realizar operaciones de detección de bordes en imágenes, tanto limpias como ruidosas, utilizando filtros basados en aproximaciones de gradientes y laplacianos.

Introducción

- Los histogramas constituyen la base de varias técnicas de procesamiento en el dominio espacial. La manipulación de los histogramas es usada de manera eficiente en el realce o mejoramiento de la calidad de una imagen. La información estadística obtenida a partir de los histogramas se utiliza en diversas aplicaciones como compresión y segmentación de imágenes. La facilidad con la que se pueden calcular los histogramas usando software y su bajo consumo de recursos de hardware en su implementación, han hecho de esta herramienta una de las más usadas en el procesamiento en tiempo real.

El histograma de una imagen es la representación gráfica de la distribución que existe de las distintas tonalidades de grises con relación al número de píxeles o porcentaje de los mismos, es decir, un histograma representa la frecuencia relativa de ocurrencia de los niveles de gris. La representación de un histograma ideal sería la de una recta horizontal, ya que eso nos indicaría que todos los posibles valores de grises están distribuidos de manera uniforme en nuestra imagen.

La ecualización del histograma es una técnica bastante conocida y sirve para obtener un histograma uniforme de tal manera que los niveles de gris son distribuidos sobre la escala y un número igual de píxeles son colocados en cada nivel de gris. Para un observador, esta ecualización hace que las imágenes se vean más balanceadas y con mejor contraste. Como consecuencia, una imagen ecualizada, permite que ciertos detalles sean visibles en regiones oscuras o brillantes.

- Los filtros espaciales tienen como objetivo modificar la contribución de determinados rangos de frecuencias de una imagen. El término espacial se refiere a que el filtro se aplica directamente a la imagen y no a una transformada de la misma, es decir, el nivel de gris de un pixel se obtiene directamente en función del valor de sus vecinos. La convolución es la operación con la cual se hace filtrado espacial. Los filtros espaciales pueden clasificarse basándose en su linealidad en filtros lineales y en filtros no lineales. A su vez los filtros lineales pueden ser clasificados según las frecuencias que dejen pasar: los filtros paso bajo atenúan o eliminan las componentes de alta frecuencia a la vez que dejan inalteradas las bajas frecuencias; los filtros paso alto atenúan o eliminan las componentes de baja frecuencia con lo que agudizan las componentes de alta frecuencia; los filtros paso banda eliminan regiones elegidas de frecuencias intermedias. A continuación se describe el uso de los diferentes filtros:

1. Filtros paso bajas: son utilizados en la reducción de ruido; suavizan y apllanan un poco las imágenes y como consecuencia se reduce o se pierde la nitidez. En inglés son conocidos como Smoothing Spatial Filters.
2. Filtros paso altas: estos filtros son utilizados para detectar cambios de luminosidad. Son utilizados en la detección de patrones como bordes o para resaltar detalles finos de una imagen. En inglés son conocidos como Sharpening Spatial Filters. Los filtros unsharp masking son filtros paso alto usados en el mejoramiento de la nitidez o de la calidad visual de una imagen.

-
3. Filtros paso banda: son utilizados para detectar patrones de ruido. Ya que un filtro paso banda generalmente elimina demasiado contenido de una imagen casi no son usados, sin embargo, los filtros paso banda son útiles para aislar los efectos de ciertas bandas de frecuencias seleccionadas sobre una imagen. De esta manera, estos filtros ayudan a simplificar el análisis de ruido, razonablemente independiente del contenido de la imagen.

Desarrollo

Resuelve los problemas de la lista siguiente y describe tu solución en cada inciso. Los incisos en donde únicamente tengas que desplegar imágenes no requieren de ninguna descripción.

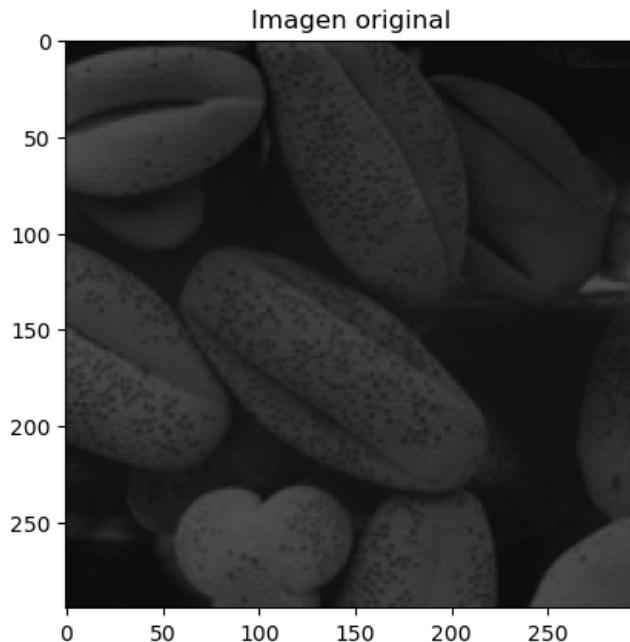
1. Aplicar a una imagen las diferentes transformaciones: negativa, logarítmica, exponencial y gama.

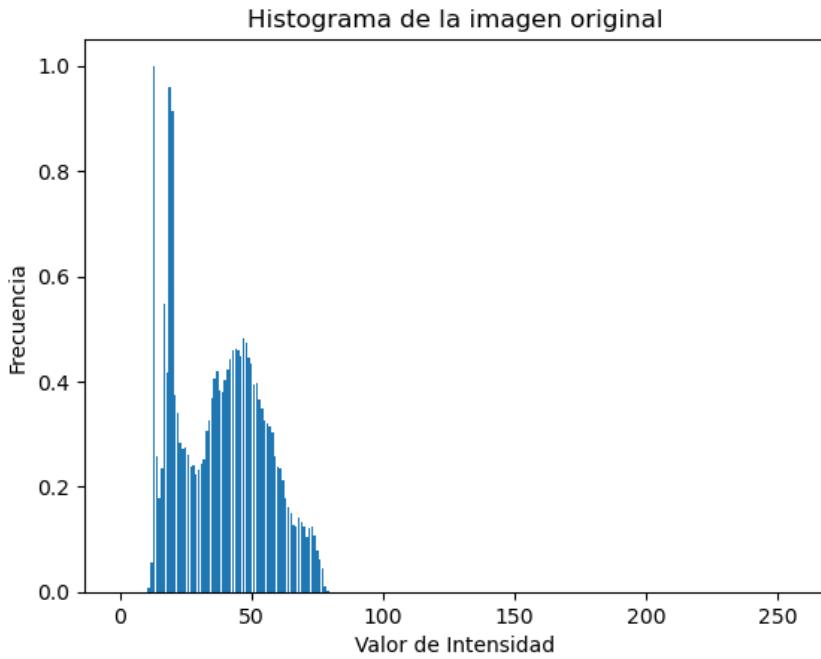
- a) Desplegar la imagen original y su histograma.

Para crear el histograma de una imagen utilizamos la función `obtener_hist()`, donde se crea un arreglo de ceros con índices de 0 a 255 que representan los diferentes valores de niveles de gris que pueden tomar los píxeles de la imagen recibida. Recorremos la imagen pixel a pixel, y vamos sumando 1 al valor del índice i si el valor del pixel es i . Como el valor del pixel se usará como índice debemos asegurarnos que sea de tipo int y que esté dentro del rango de índices.

Por otra parte tenemos la función `obtener_hist_eq()` que utiliza la función anterior para obtener el histograma de la imagen recibida. A cada uno de los valores del arreglo lo dividimos entre el valor máximo entre los tonos para ecualizar el histograma y obtener valores entre 0 y 1.

Definido lo anterior, la imagen recibida y su histograma ecualizado se despliegan de la siguiente manera:



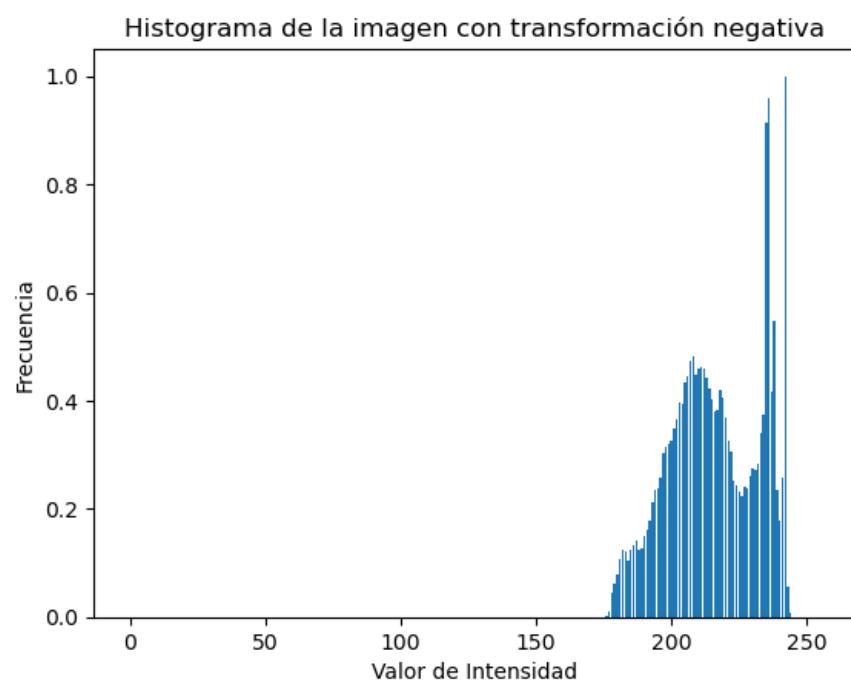
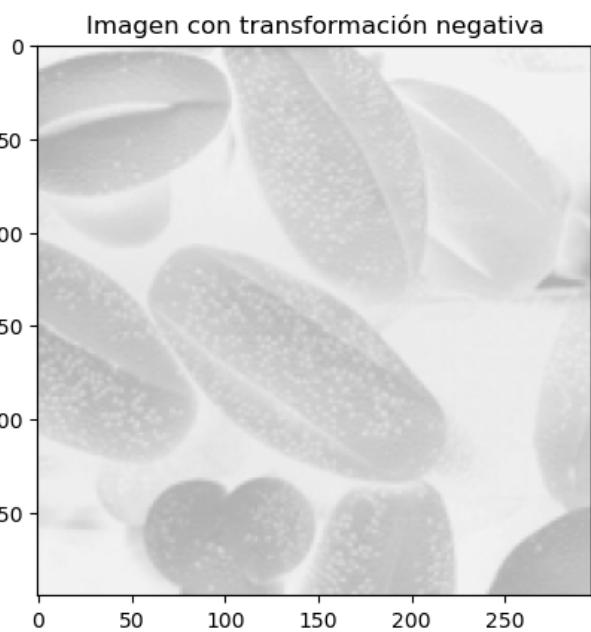


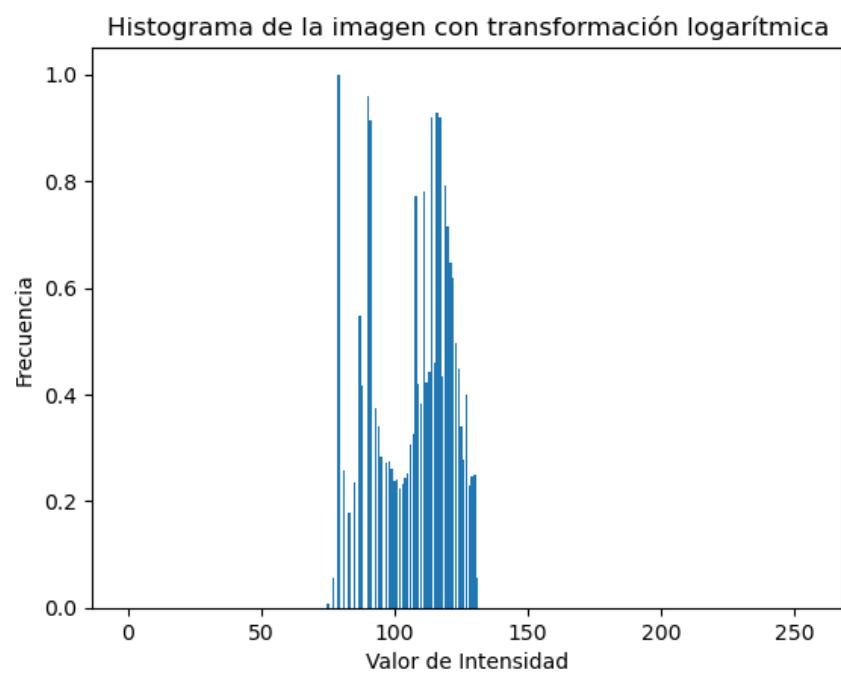
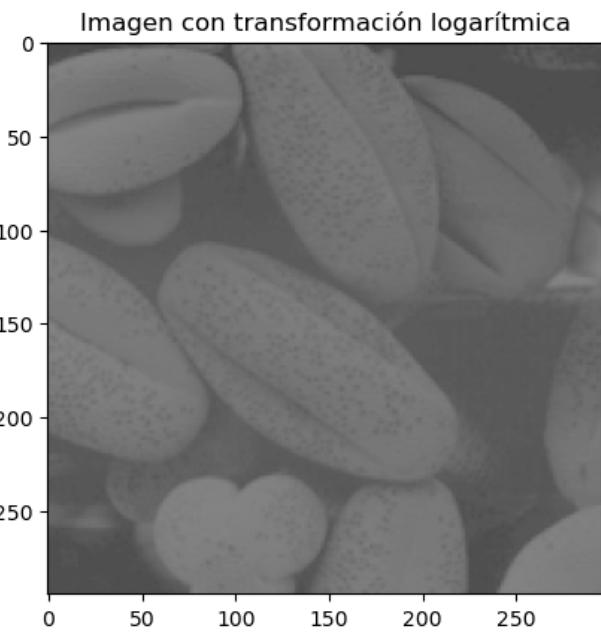
- b) Aplicar cada una de las transformaciones referidas a la imagen leída en (a) y desplegar en la misma ventana la imagen resultado con su respectivo histograma.

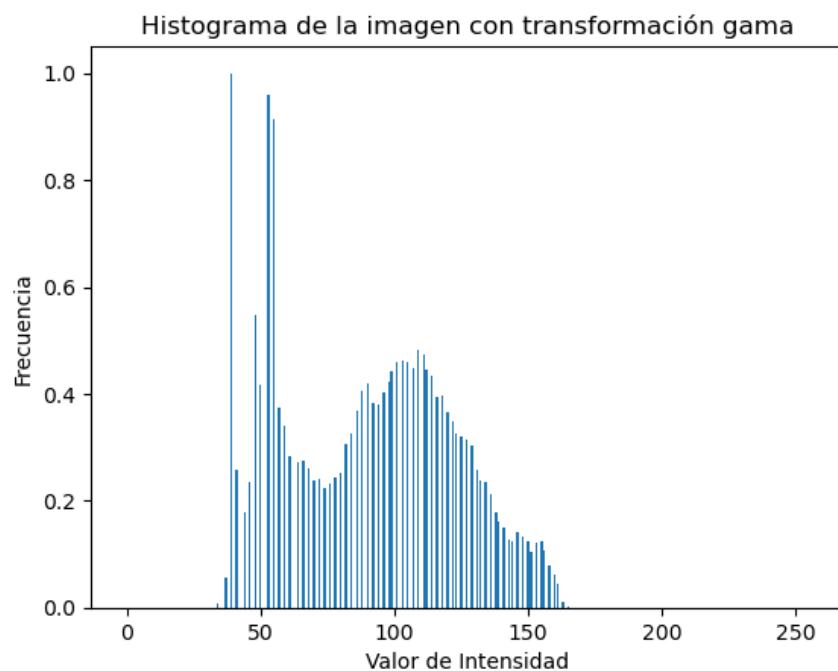
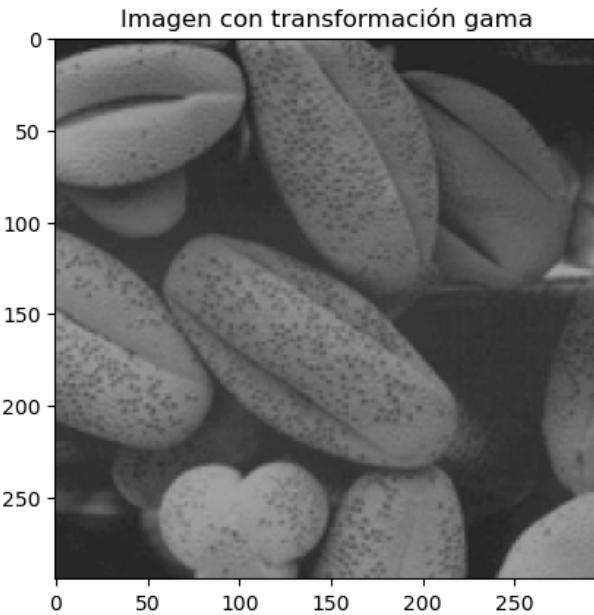
Para aplicar las diferentes trasformaciones se creó una función para cada una de ellas, las cuáles reciben una imagen y variables extra si se requiere. Todas inician creando una nueva imagen llena de ceros (negra) del tamaño de la imagen recibida, esta imagen es procesada pixel a pixel:

- Para la función `negativa()` se le asigna al pixel $imagen_nueva[x][y]$ el valor $255 - imagen_original[x][y]$, para obtener su correspondiente valor en negativo de la imagen original.
- Para la función `logaritmica()` se le asigna al pixel $imagen_nueva[x][y]$ el valor $c * \log(1 + imagen_original[x][y])$, para obtener su correspondiente valor logarítmico de la imagen original. La constante c puede ser recibida como parámetro, pero por defecto tiene valor de 1. Adicionalmente, por el tipo de operación que se realiza sobre el pixel este puede resultar en un número decimal, por lo que el valor es redondeado.
- Para la función `gama()` se le asigna al pixel $imagen_nueva[x][y]$ el valor $c * (imagen_original[x][y])^{gama}$, para obtener su correspondiente valor exponencial de la imagen original. La variable $gama$ y la constante c pueden ser recibidas como parámetro, pero por defecto tienen valor de 0.4 y 1 respectivamente. Adicionalmente, por el tipo de operación que se realiza sobre el pixel este puede resultar en un número decimal, por lo que el valor es redondeado.

Para todas las funciones, una vez recorridos y procesados todos los píxeles podemos regresar la imagen creada.





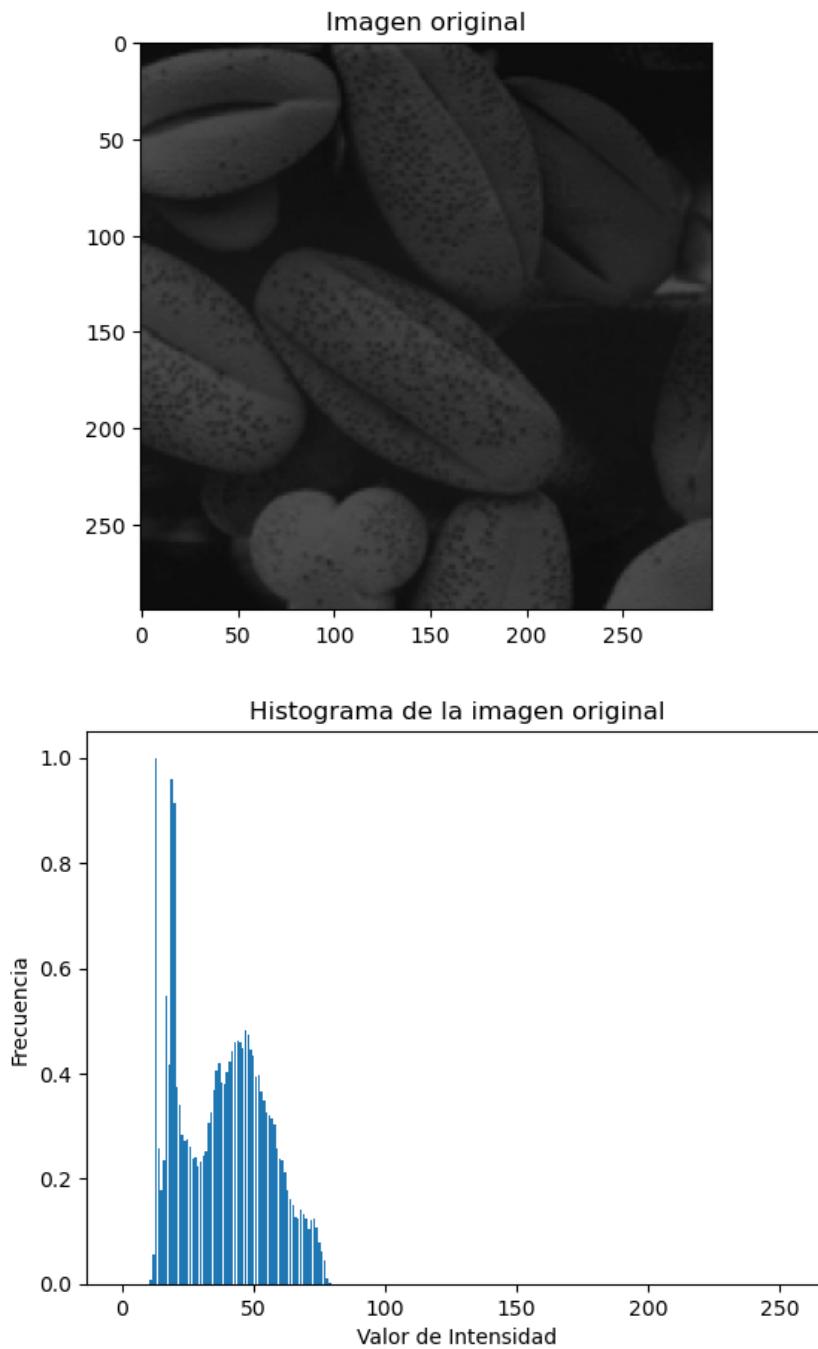


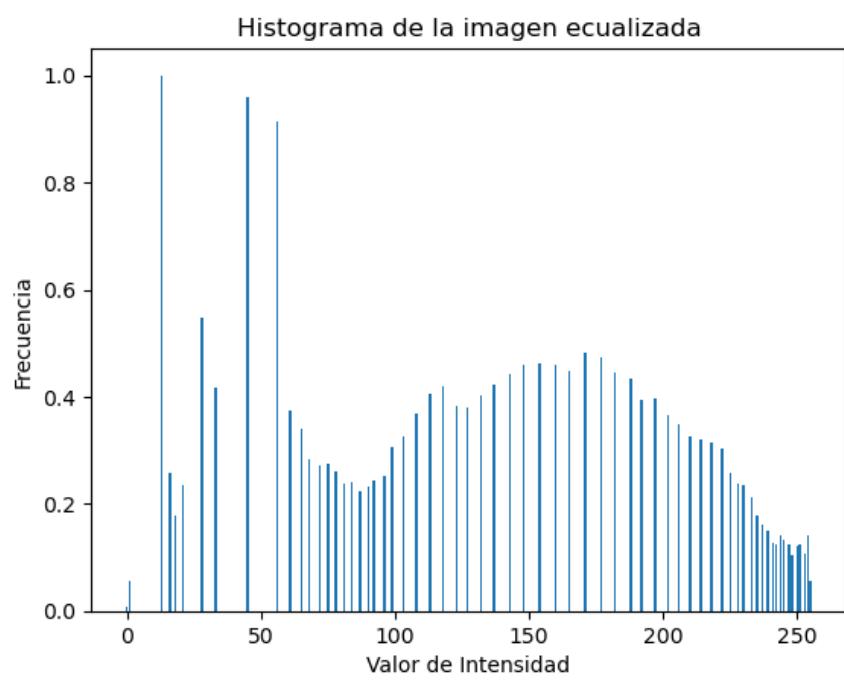
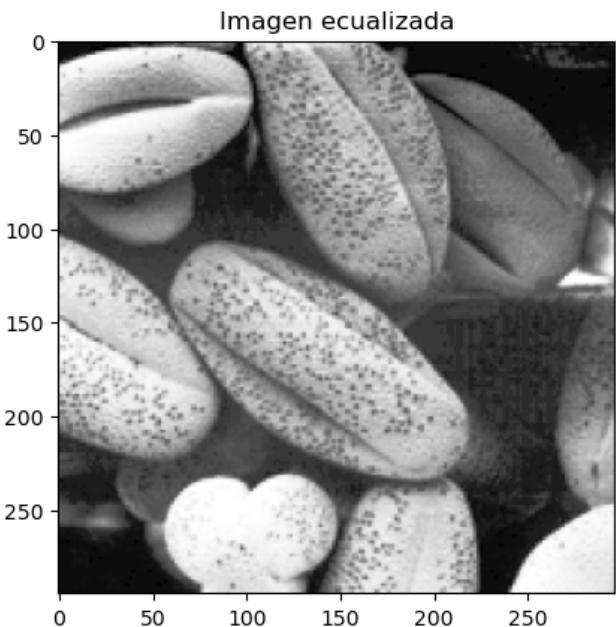
- c) Ecuallizar el histograma de una imagen. Desplegar en la misma ventana: imagen original, su histograma, imagen ecualizada y su histograma.

Creamos la función `ecualizacion_img()` que a partir del histograma de la imagen recibida, aplica la función de distribución acumulada sobre los tonos de gris. Ya que tenemos probabilidades acumuladas que suman 1, vamos a volver a dimensionarlos multiplicando cada una por 255 y redondeando su valor.

Creamos una imagen llena de ceros con el mismo tamaño que la original, y a cada uno de sus píxeles se le asigna el valor que le corresponde en el arreglo de probabilidades acumuladas según el valor

de ese pixel en la imagen original. Una vez procesados todos los pixeles podemos regresar nuestra imagen ecualizada.





-
- Aplicar a una imagen sin ruido y la misma imagen con ruido filtros de suavizamiento y realce. La imagen con ruido se puede generar a partir de la imagen sin ruido.

Para generar las imágenes con ruido se definieron dos funciones: `ruido_gaussiano()` y `ruido_sal_pim()`.

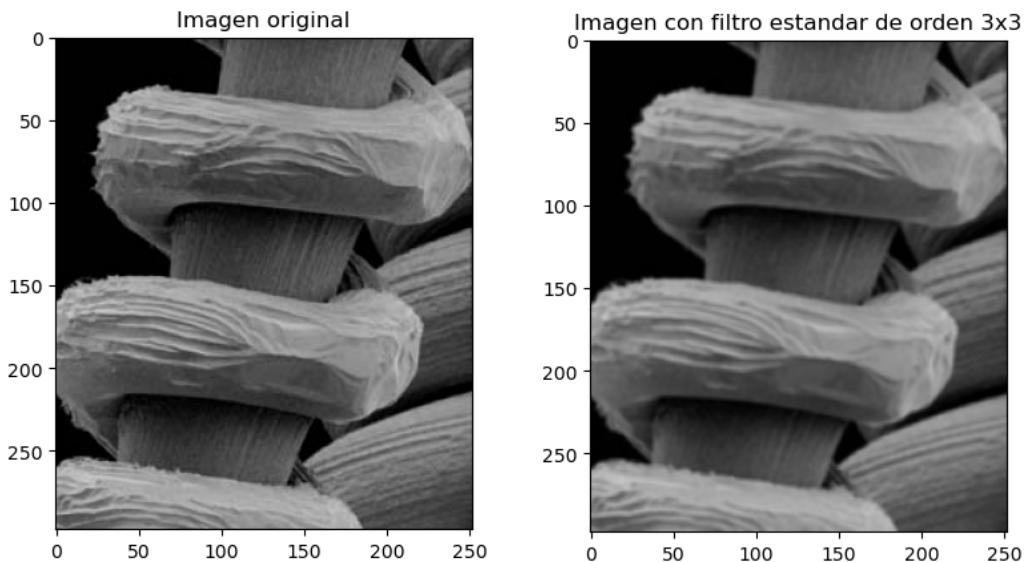
El ruido gaussiano se genera con ayuda de la función `np.random.normal()` que regresa una matriz de números aleatorios del mismo tamaño que la imagen, donde los números seguirán una distribución normal (gaussiana) con la media y la desviación estándar especificados.

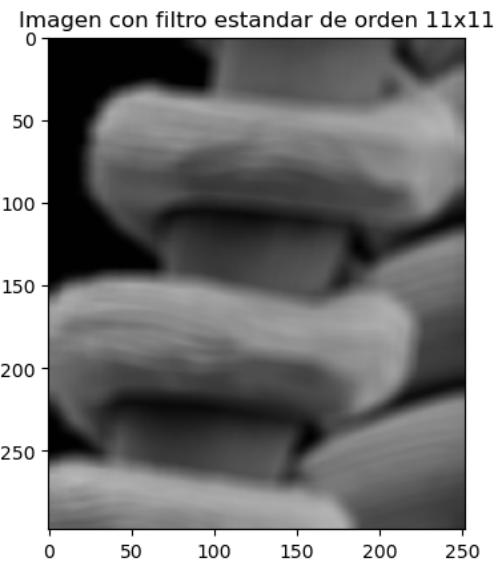
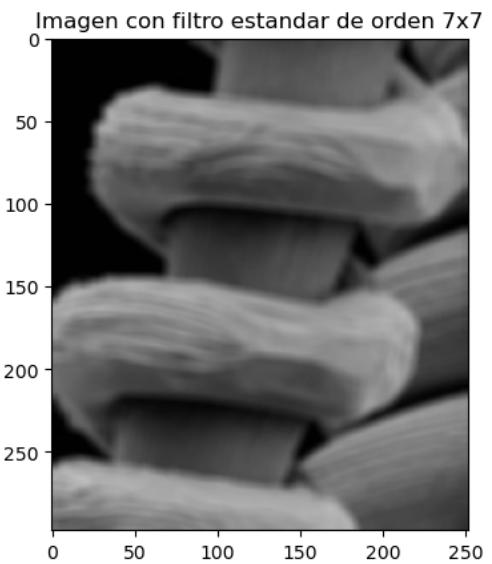
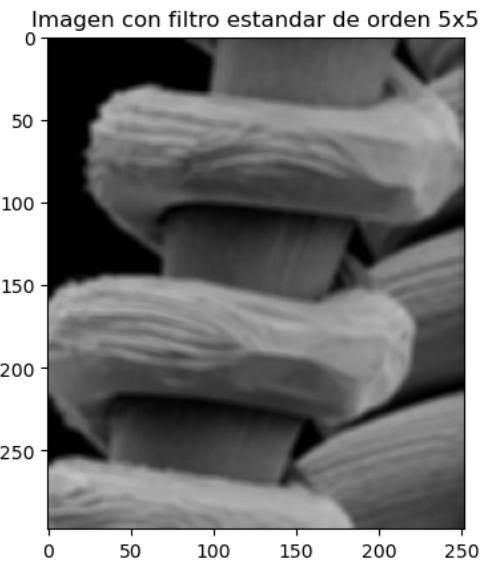
Para el ruido sal y pimienta, a partir de un porcentaje especificado, la mitad de ese porcentaje serán píxeles con valor de cero (negro) y la otra mitad con valor de 255 (blanco), todos ellos con coordenadas aleatorias.

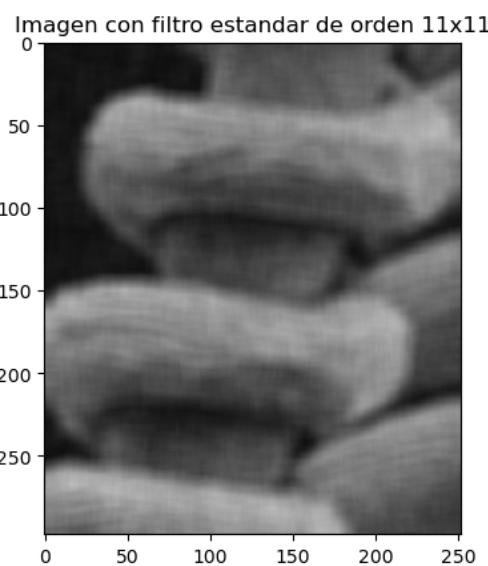
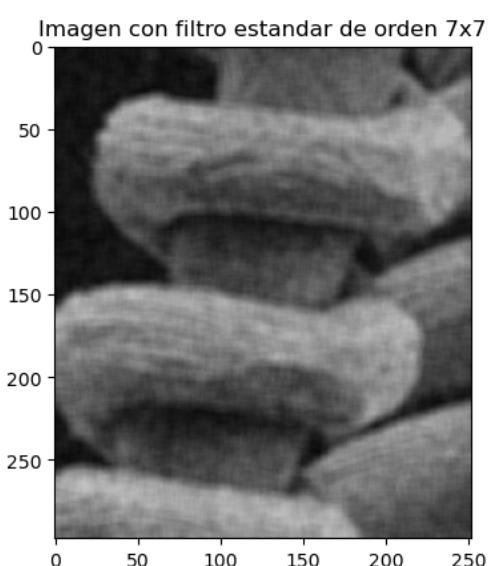
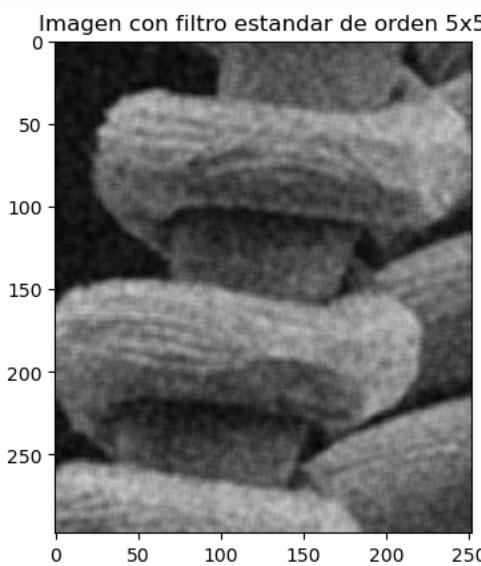
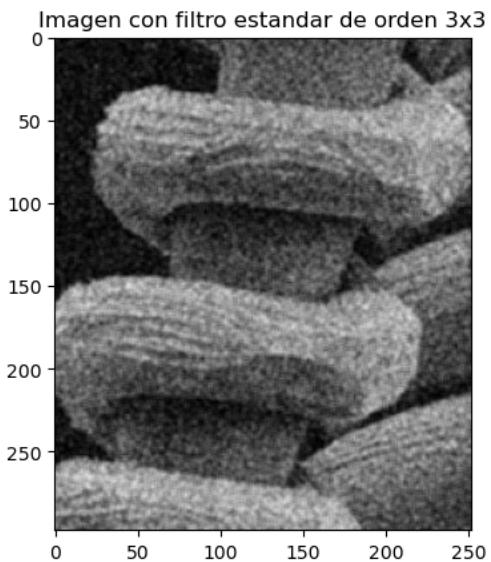
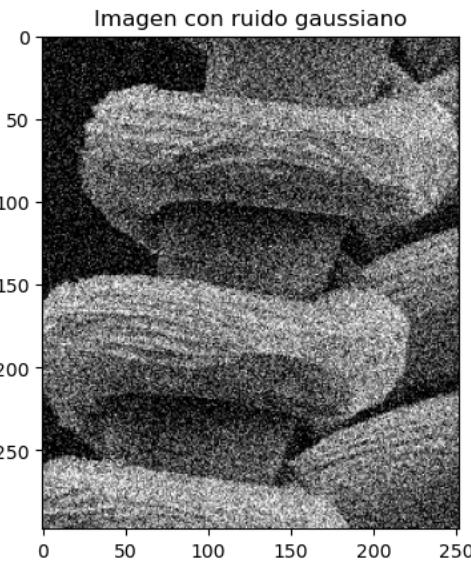
Para aplicar todos los filtros se ocupará una función de convolución, `convolucion()`, que recibe una imagen, las coordenadas x y y de uno de sus píxeles y la máscara que le será aplicada. Se recorren los píxeles de la imagen que son cubiertos por la máscara si su centro se posiciona en las coordenadas recibidas. Verificamos que existan dichas coordenadas en la imagen y almacenamos el producto del valor de la imagen y la máscara en esa posición en una lista. Finalmente se regresa la lista de valores generados.

- Aplicar los filtros paso bajas promedio estándar a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 5x5, 7x7 y 11x11.

La función para el filtro paso bajas promedio estándar, `filtro_estandar()`, crea una máscara de unos del orden que sea indicado y la cual será aplicada en la convolución. Al recorrer pixel por pixel se aplica la convolución, se suman los elementos que devuelva y se divide entre el tamaño de la máscara, este es el valor que le será asignado al pixel correspondiente en la nueva imagen. Debido a que se hace una división aplicamos `round` para asegurar que no se queden números decimales.







-
- b) Aplicar los filtros paso bajas promedio ponderado a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 5x5, 7x7 y 11x11.

Se ocupa una función `mascara_dist` para crear una máscara del orden recibido, que se base en la distancia (`city_block`) de cada pixel al pixel del centro tal que mientras más cerca a él mayor sea su peso.

La función para el filtro paso bajas promedio ponderado, `filtro_ponderado()`, crea una máscara con la función anterior, recorre la imagen, aplicando la convolución del pixel con la máscara, sumando la lista resultante y dividiéndolo entre la suma de elementos en la máscara. Debido a que se hace una división aplicamos `round` para asegurar que no se queden números decimales.

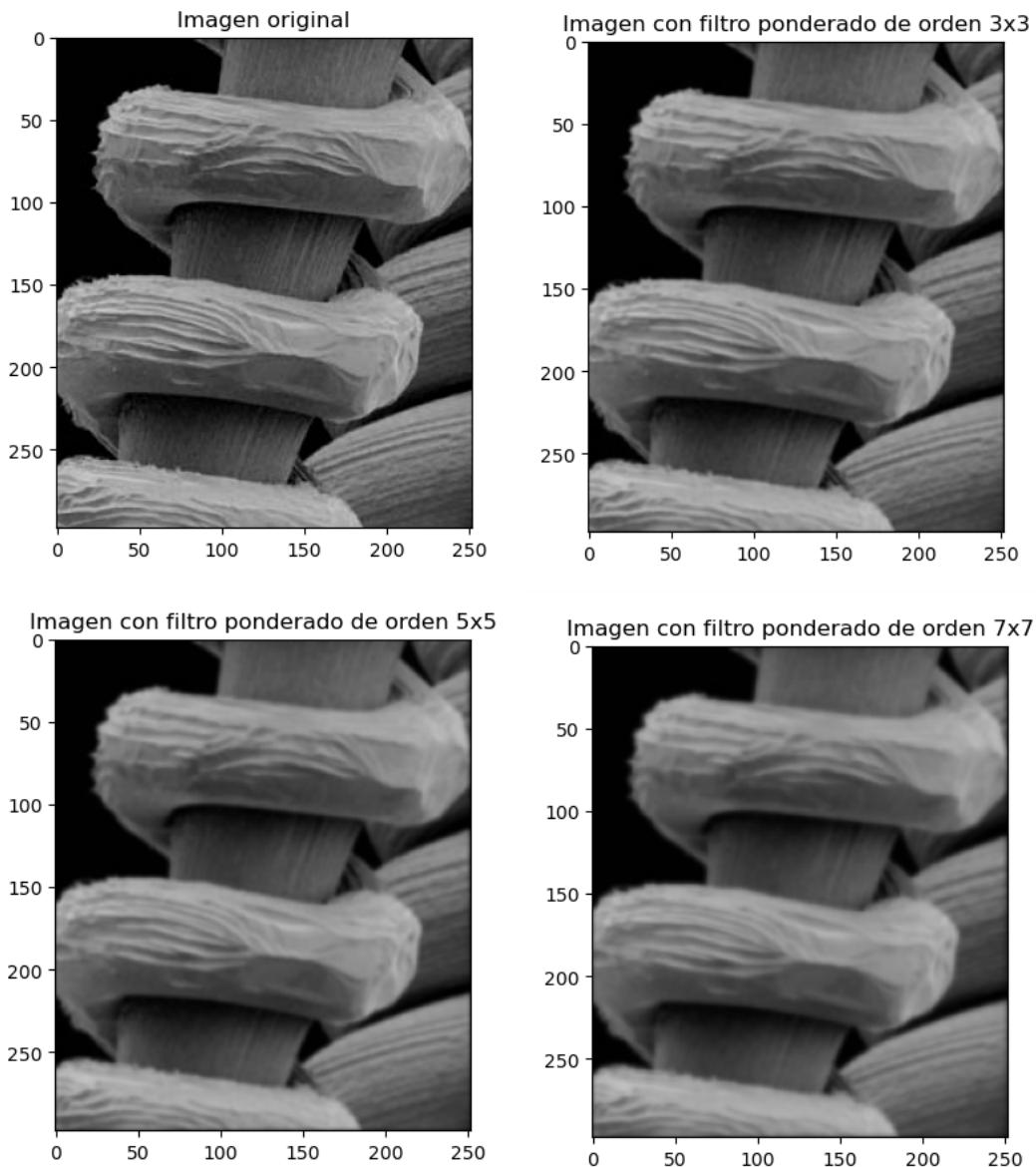


Imagen con filtro ponderado de orden 11x11

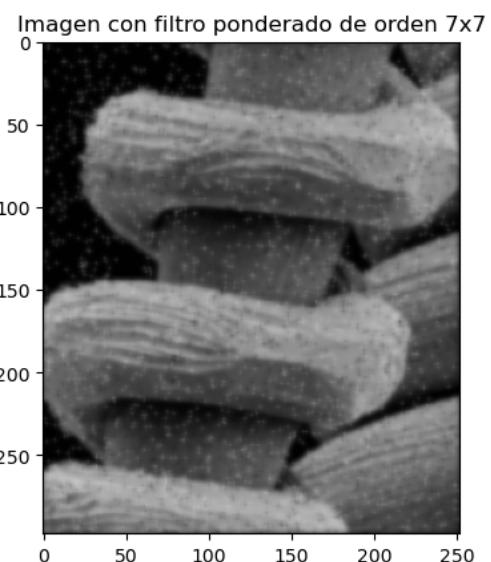
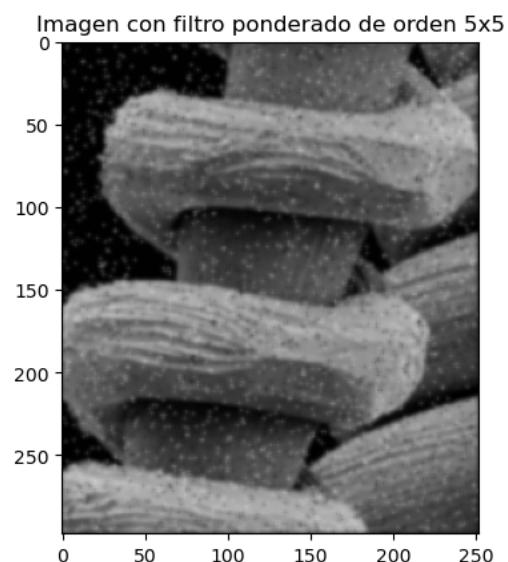
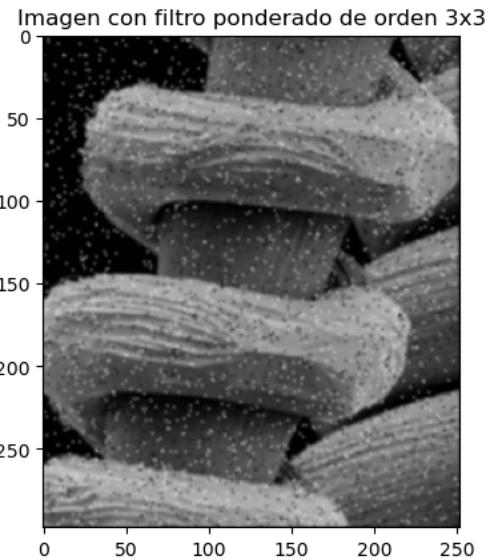
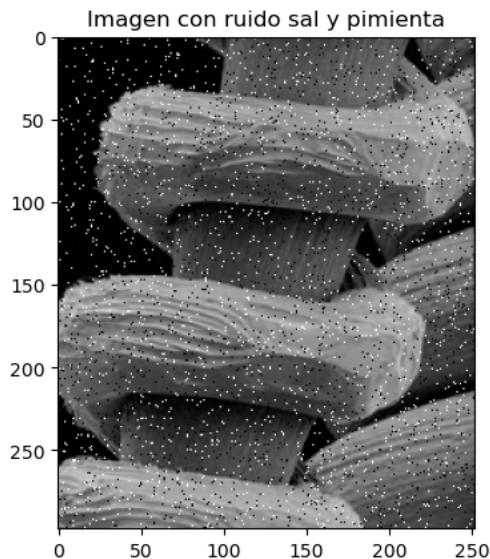
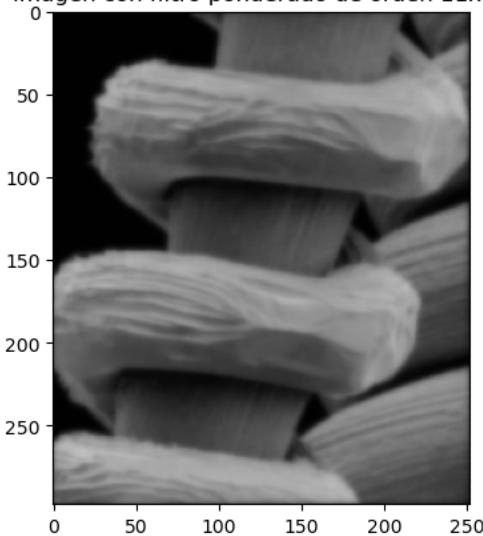
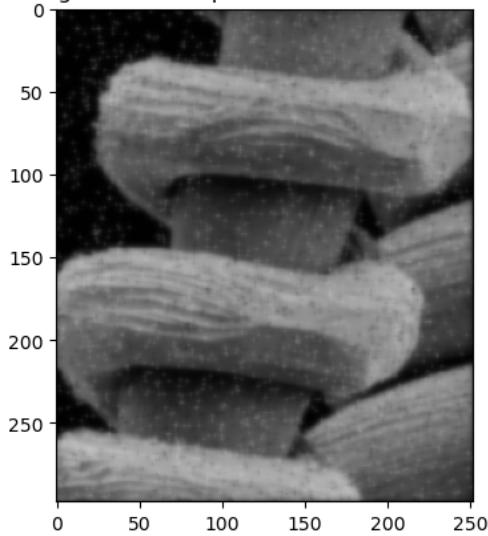


Imagen con filtro ponderado de orden 11x11



- c) Aplicar a la imagen con ruido el filtro mediana de 3x3, 5x5, 7x7 y 11x11. Utilizar ruido 'sal y pimienta' y 'gaussiano'. Compararlos.

Para el filtro mediana tenemos la función `filtro_mediana()` que realiza la convolución con una máscara de unos del orden recibido, pero no se realiza la suma de los productos, simplemente vamos a ordenar los elementos que devuelve y asignar el valor de en medio al nuevo pixel.

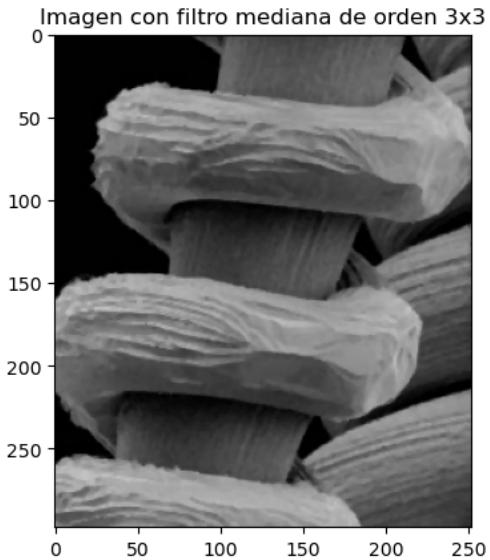
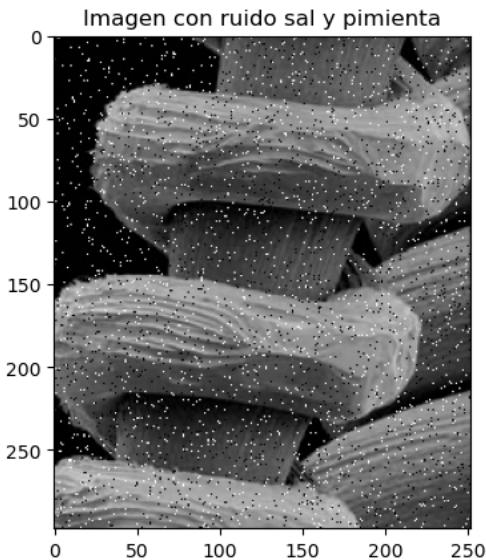


Imagen con filtro mediana de orden 5x5

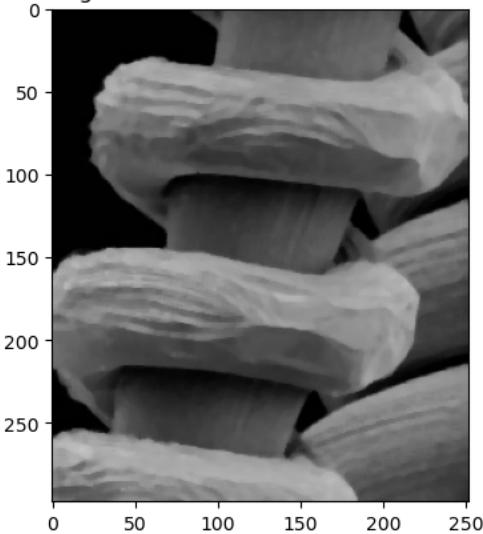


Imagen con filtro mediana de orden 7x7

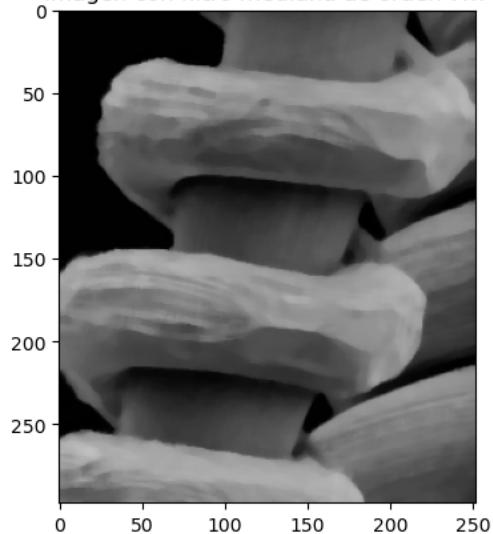


Imagen con filtro mediana de orden 11x11

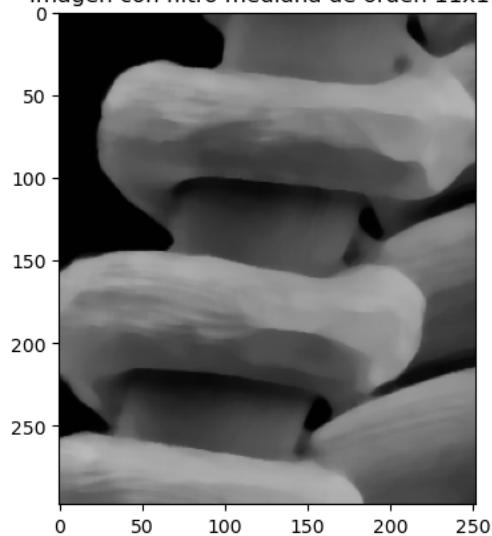


Imagen con ruido gaussiano

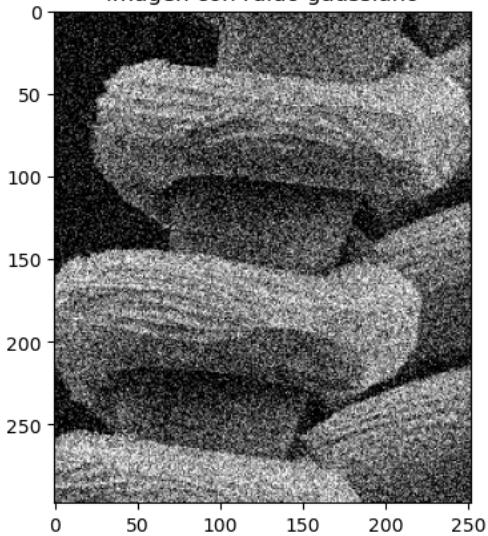
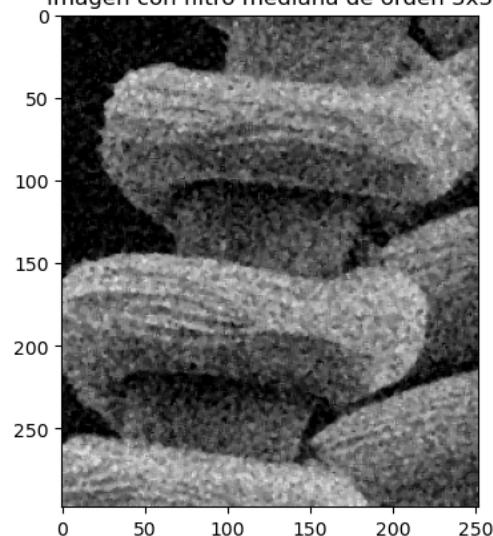
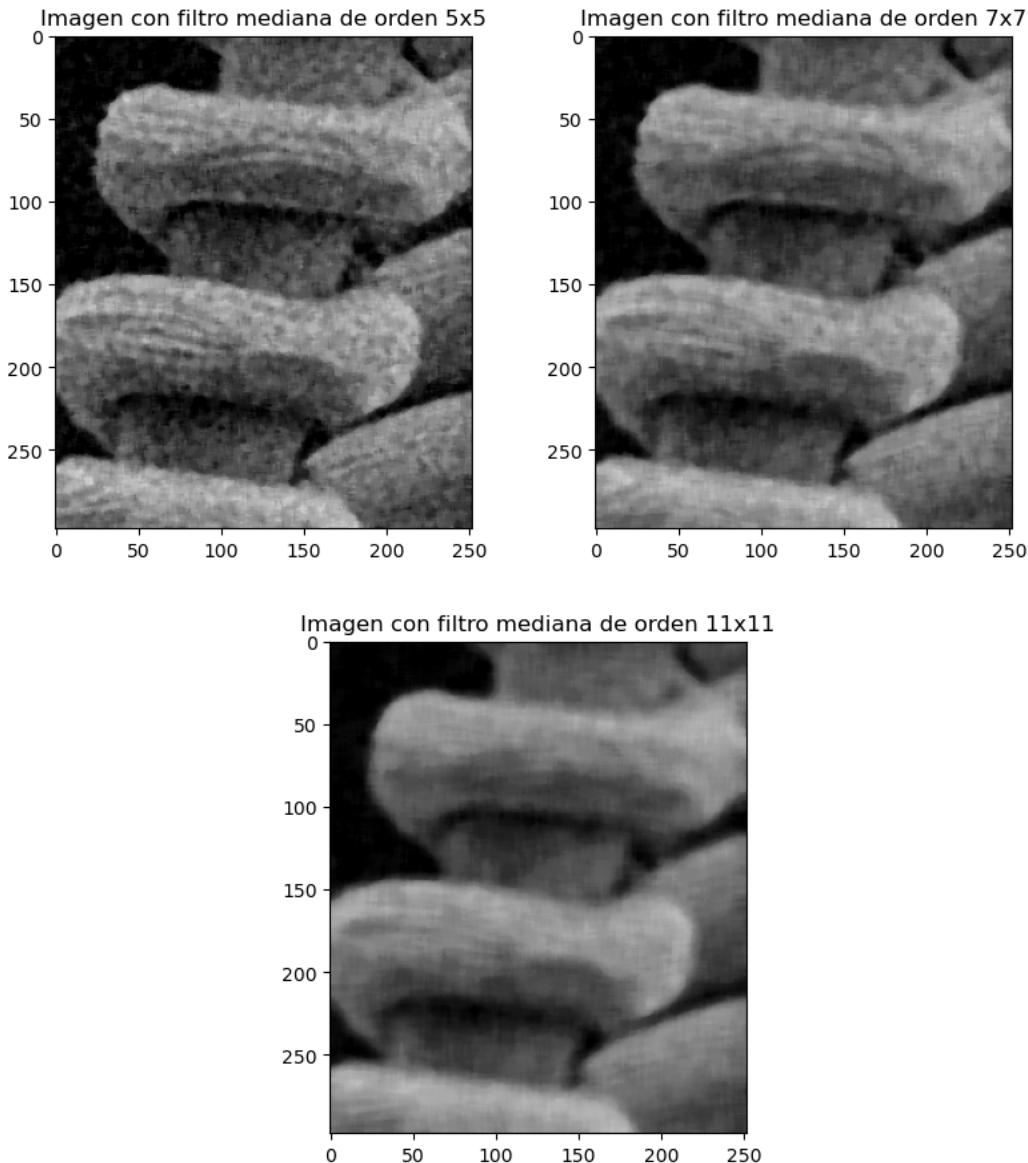


Imagen con filtro mediana de orden 3x3





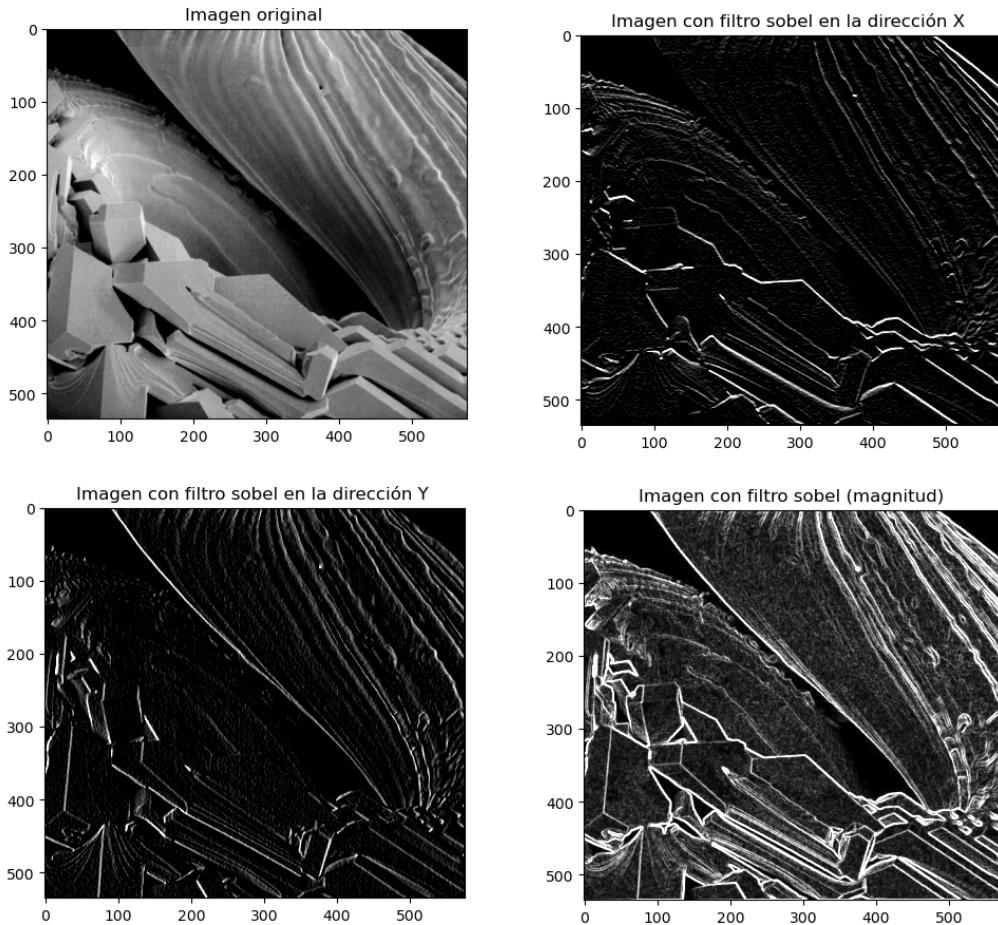
Notamos que el filtro mediana trabaja considerablemente bien eliminando el ruido sal y pimienta, ya que desde su primer aplicación puede eliminarlo en su totalidad o casi en su totalidad.

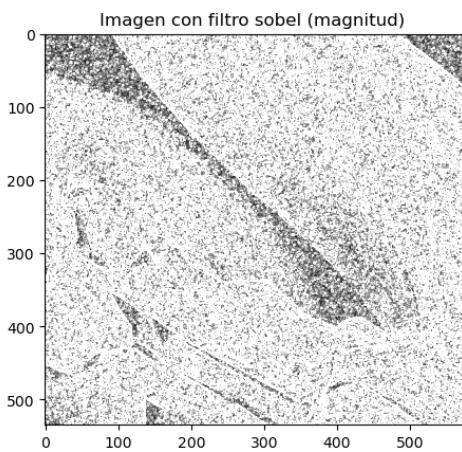
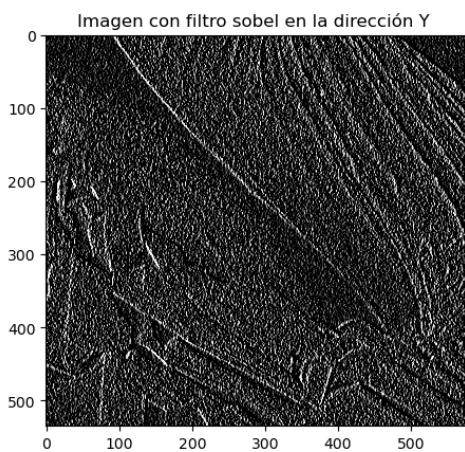
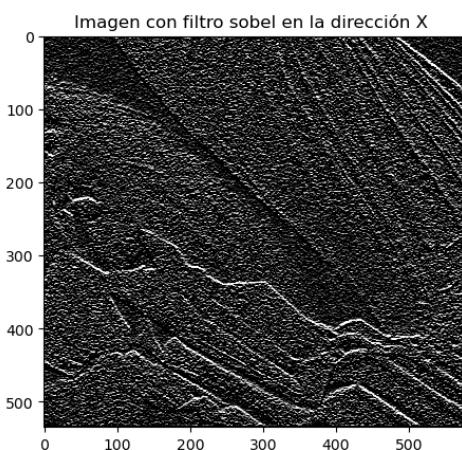
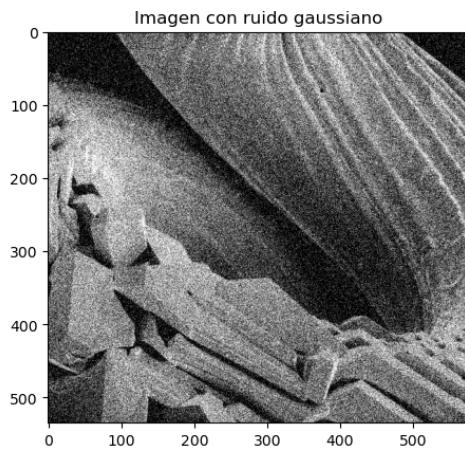
A diferencia del ruido gaussiano, con el que sigue estando presente en la imagen hasta que la imagen ya esta suficientemente difuminada para poder distinguirlo pero tambien cuesta distinguir la imagen.

- d) Aplicar a la imagen sin ruido y con ruido los filtros basados en la primera derivada o detectores de borde siguientes:
- Prewitt en la dirección X y en la dirección Y, así como su magnitud.

En la función `filtro_prewitt()` creamos tres imágenes de ceros del tamaño de la imagen recibida, uno corresponderá al gradiente en x , otro al gradiente en y y el último para la magnitud. Definimos una máscara para el gradiente en x y otra para el gradiente en y . Aplicamos la suma de la convolución a cada pixel de la imagen, una por cada máscara y guardaremos cada una en

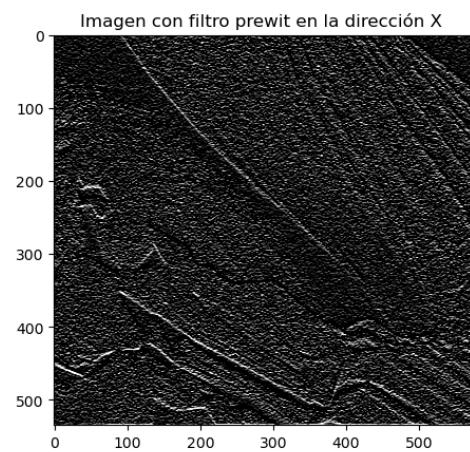
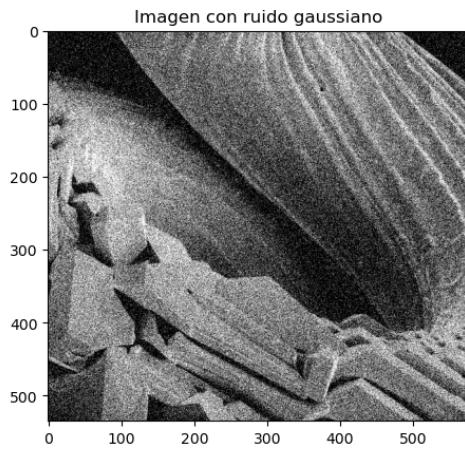
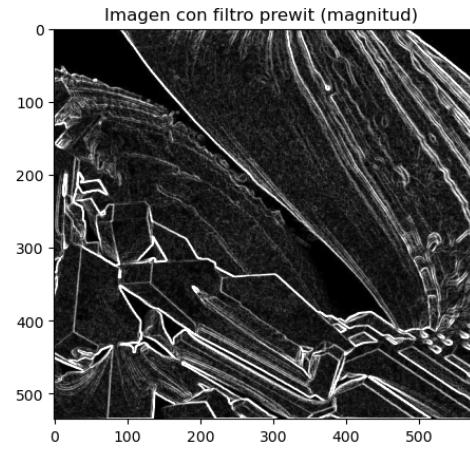
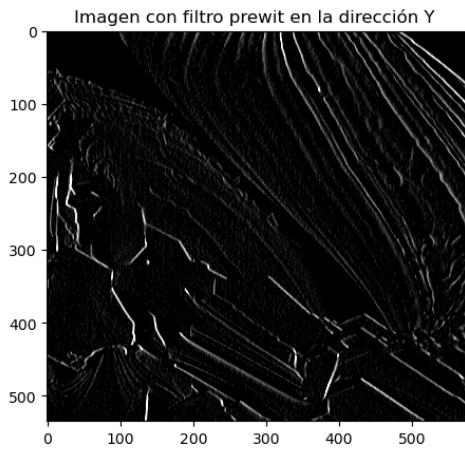
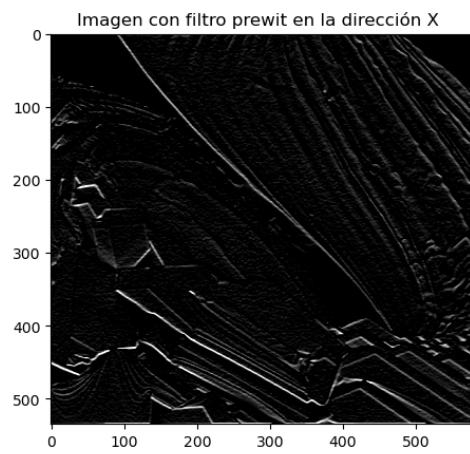
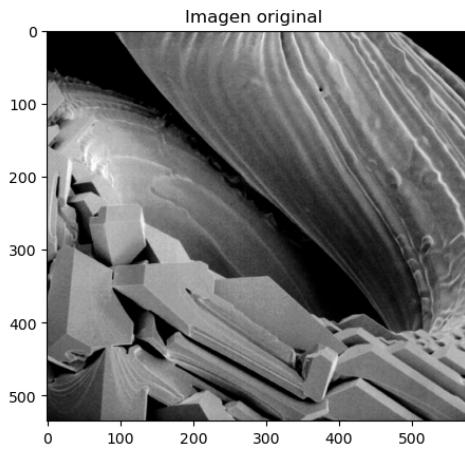
el pixel de su respectiva imagen, para la magnitud se suman los valores absolutos de ambos. Una vez procesados todos los pixeles podemos regresar nuestra lista con las 3 imágenes.

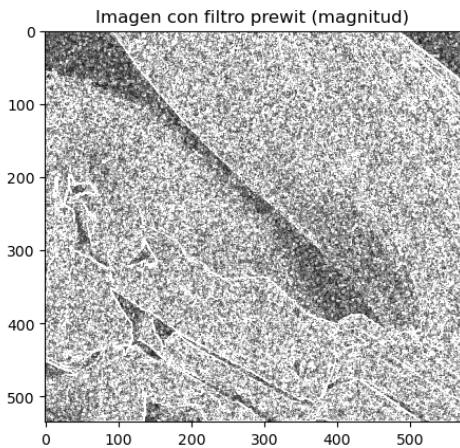
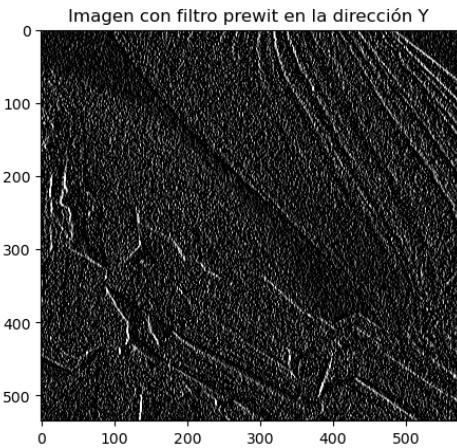




- Sobel en la dirección X y en la dirección Y, así como su magnitud.

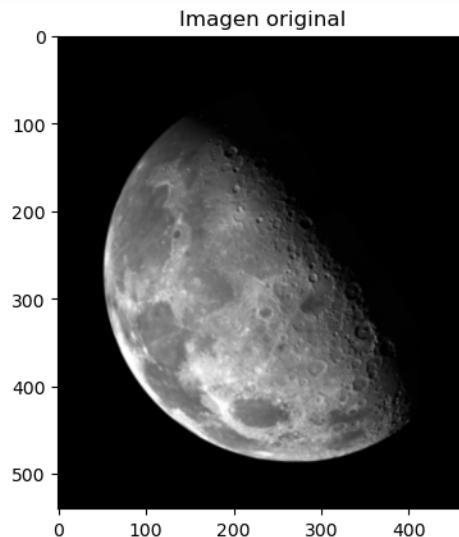
La función `filtro_sobel()` es similar a la de prewitt con la diferencia que se crean otras máscaras para el gradiente en x y otra para el gradiente en y , las correspondientes al filtro sobel.

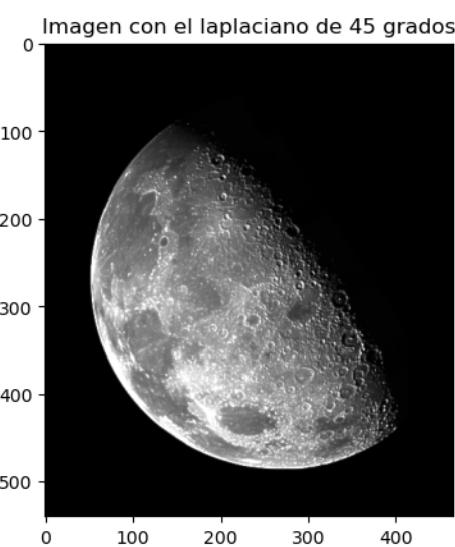
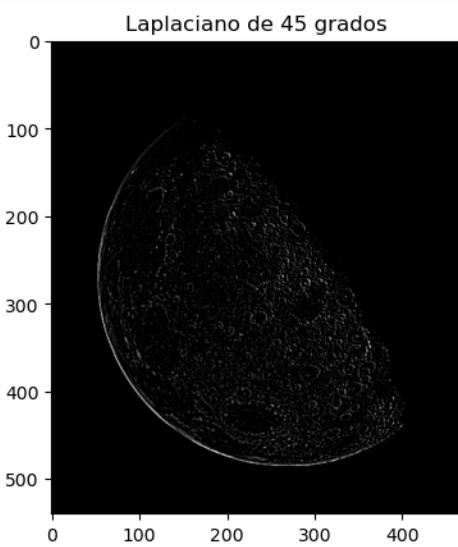
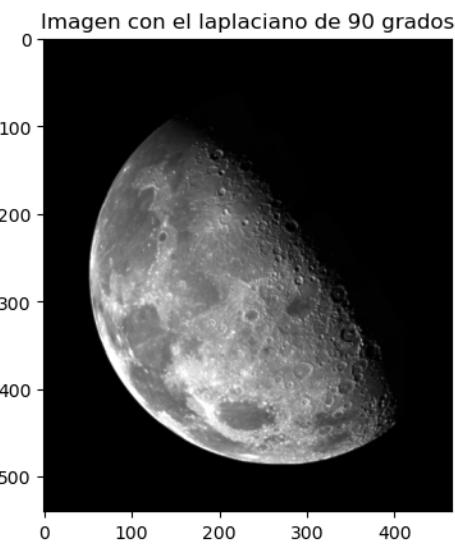
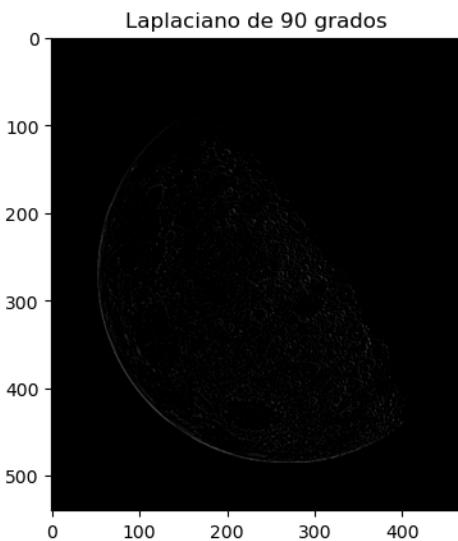




- e) Aplicar a una imagen el realce basado en las segundas derivadas (Laplaciano). Isotrópicos a 45 grados así como a 90 grados.

Para la función `laplaciano()` se crean dos máscaras para calcular el laplaciano, una isotrópica a 45 grados (si se recibe como parámetro `grado = 45`) y otra isotrópica a 90 grados (en este caso es el valor por defecto del grado, así que no es necesario recibirlo como parámetro). Se aplica la convolución con la máscara seleccionada, se suman y asignan al respectivo pixel de la nueva imagen. Regresamos una lista con la imagen del laplaciano y otra del laplaciano aplicado a la imagen (la suma).



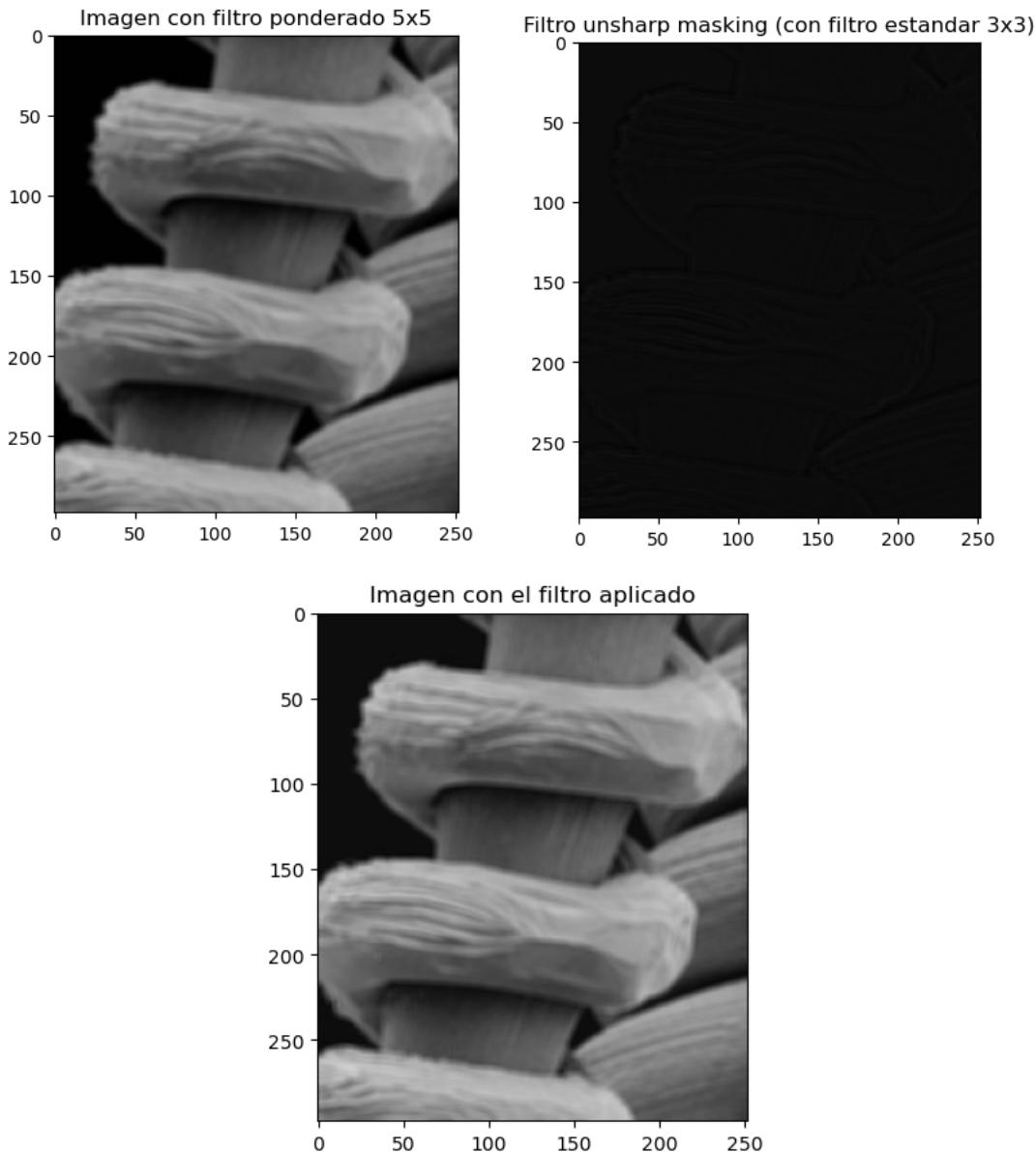


-
- f) Difuminar las imágenes sin ruido y con ruido usando un filtro paso bajas de orden 5x5, de tal manera que se obtenga una imagen sin ruido y con pérdida de nitidez y otra imagen con ruido y perdida de nitidez. Para cada uno de los siguientes incisos, filtrar las imágenes utilizando el filtro unsharp masking encontrado con los siguientes tipos de filtro paso bajas:

En la función `resta_imagenes` se reciben dos imágenes y las resta pixel a pixel. Dentro de ella reescalamos los valores para que puedan ser desplegados aquellos valores que queden fuera del rango.

Finalmente para `filtro_unsharp_masking` restamos a la primer imagen recibida su versión emborronada. Regresamos una lista con la imagen del filtro unsharp y otra de la imagen original más la anterior.

- Filtro paso bajas promedio estándar de orden 3x3 y 7x7.



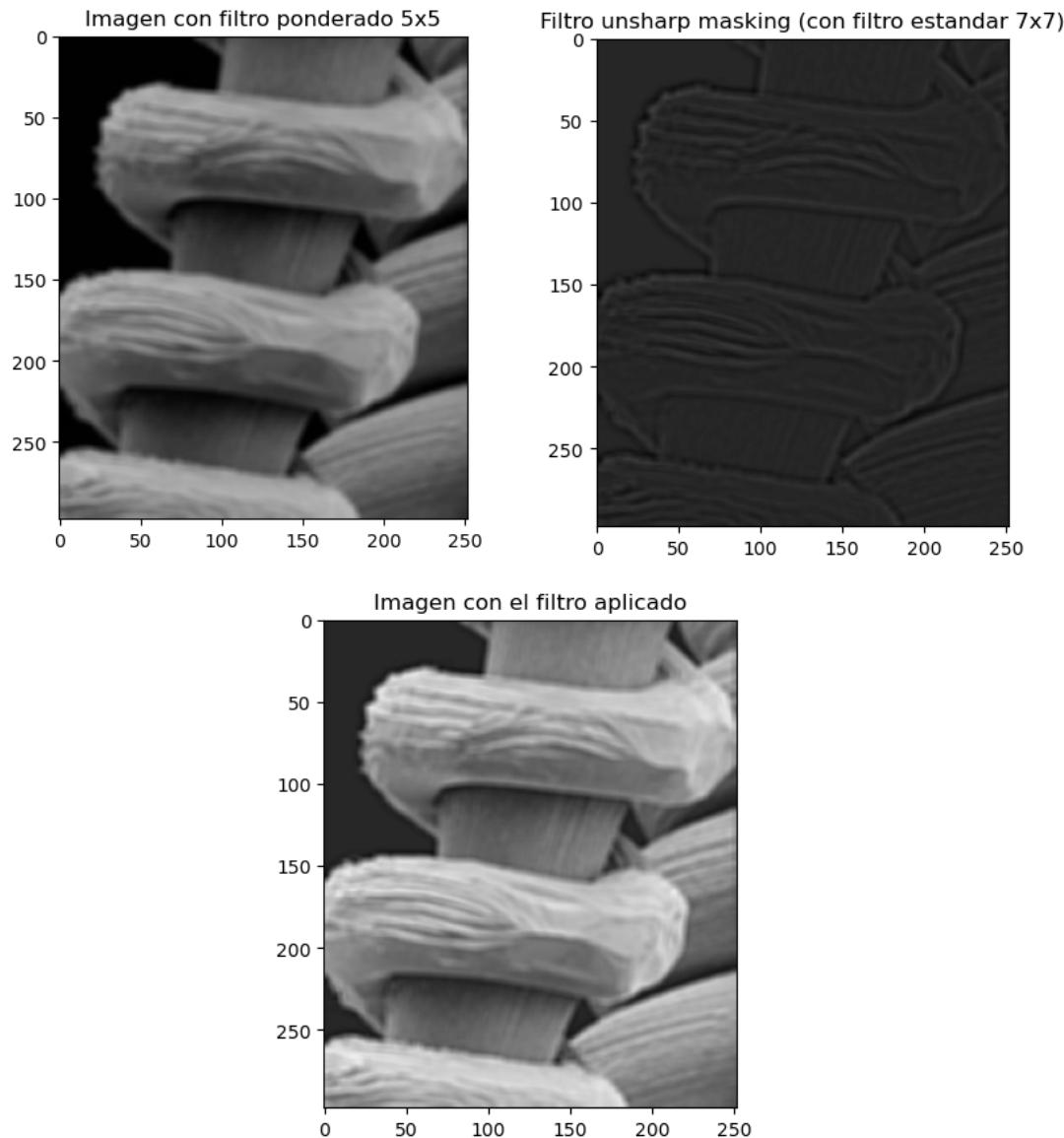
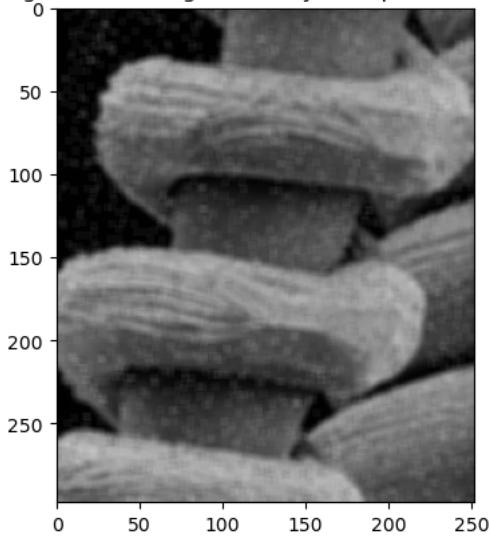


Imagen con ruido gaussiano y filtro ponderado 5x5



Filtro unsharp masking (con filtro estandar 3x3)

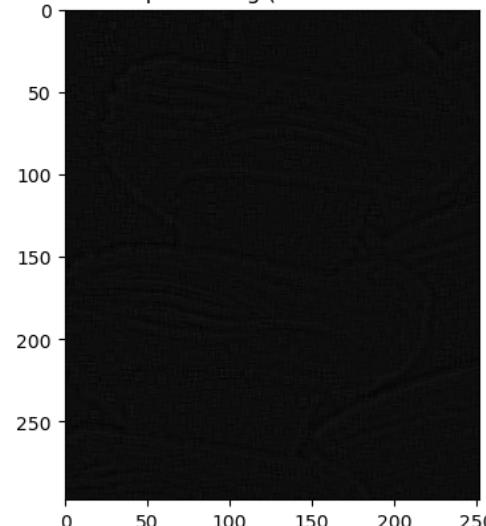
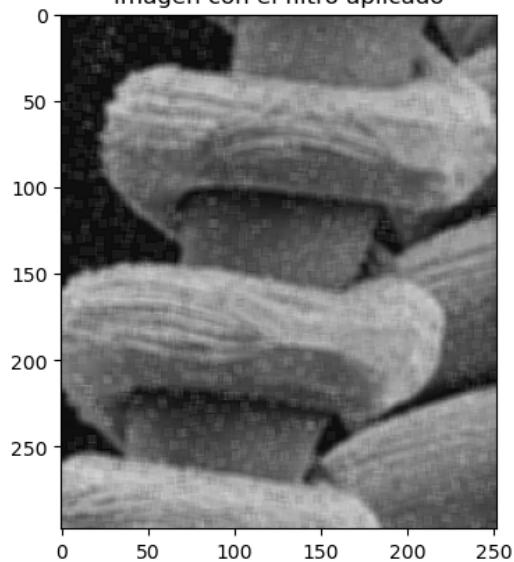
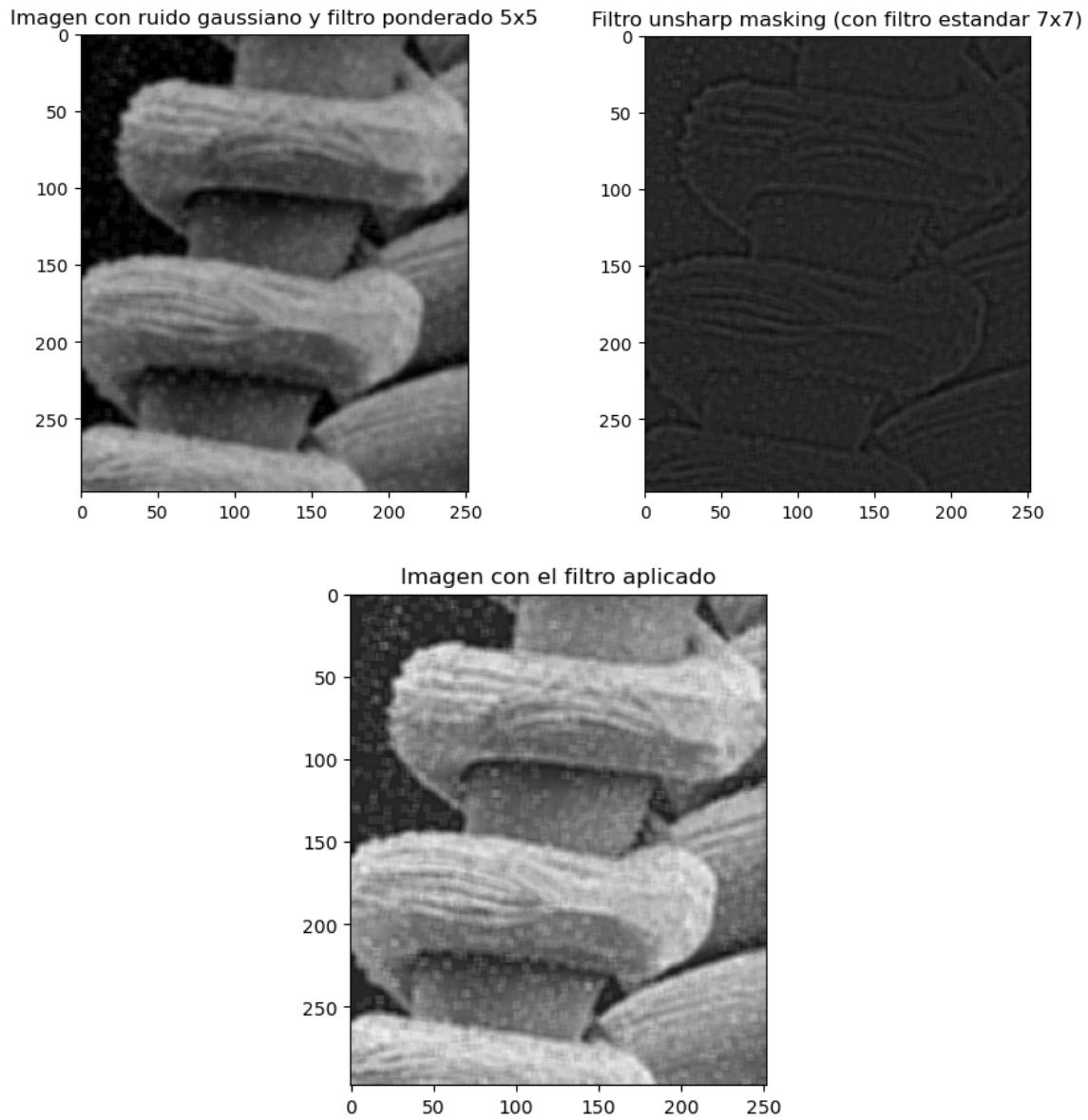
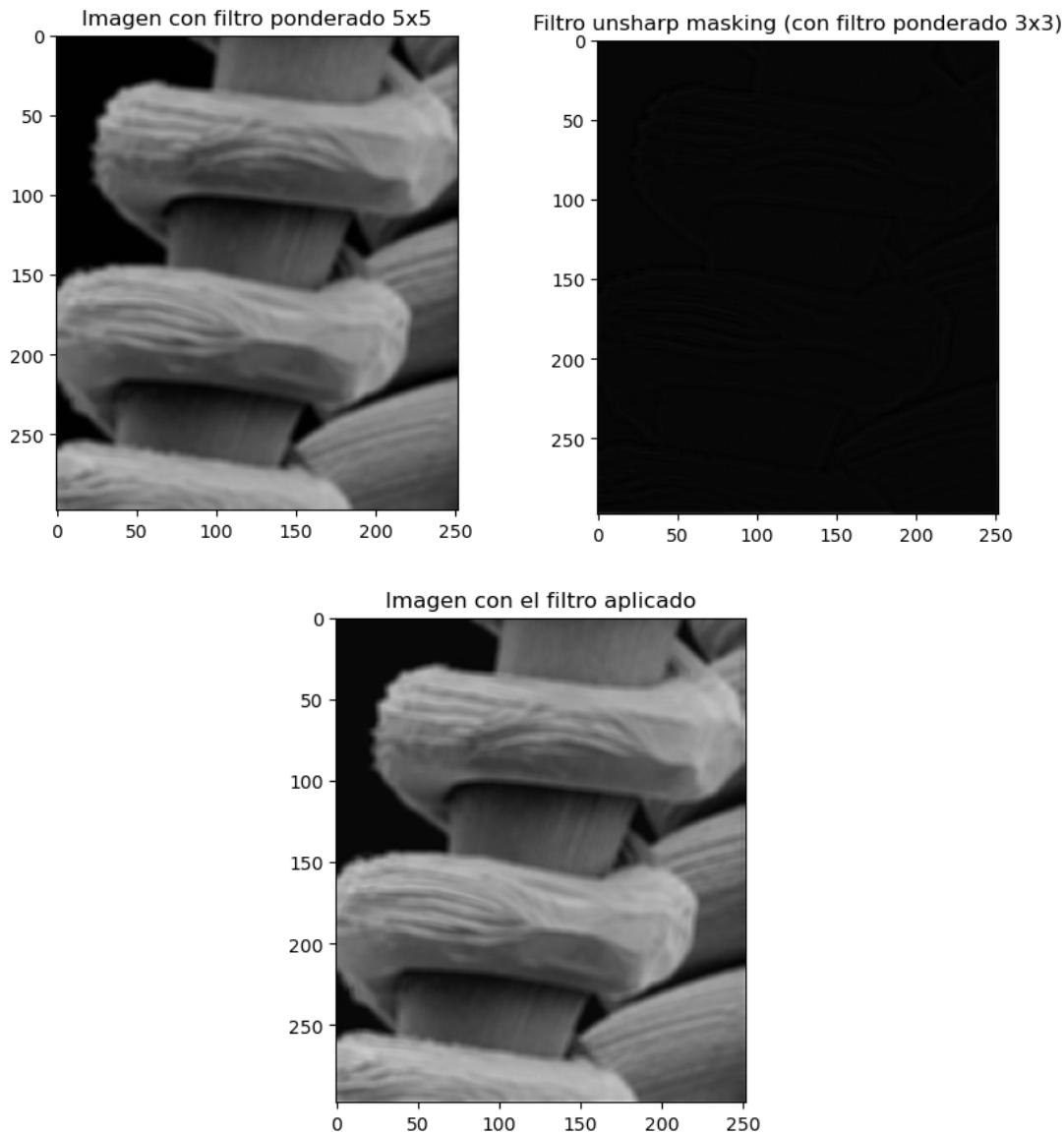


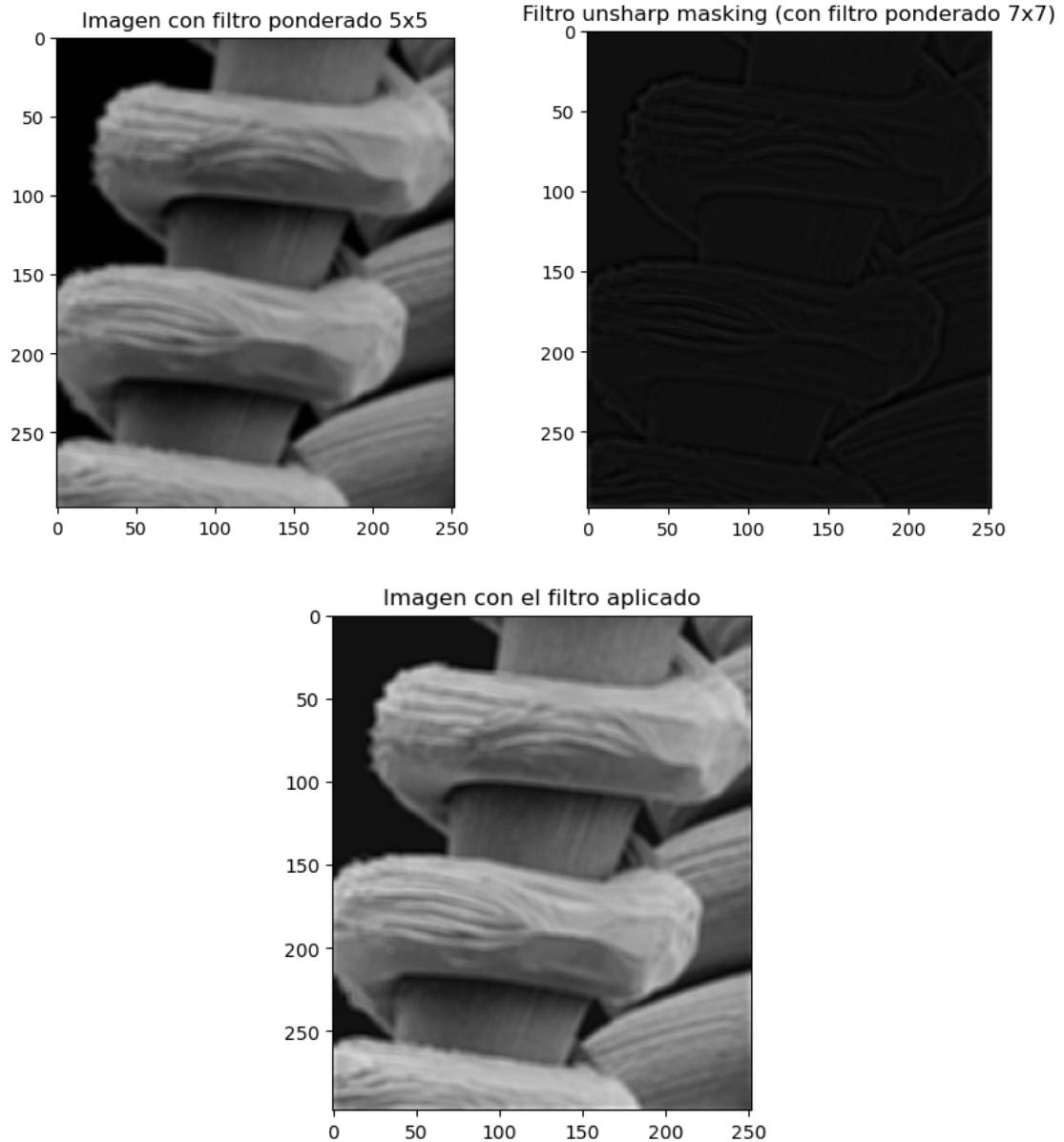
Imagen con el filtro aplicado





-
- Filtro paso bajas promedio ponderado de orden 3x3 y 7x7.





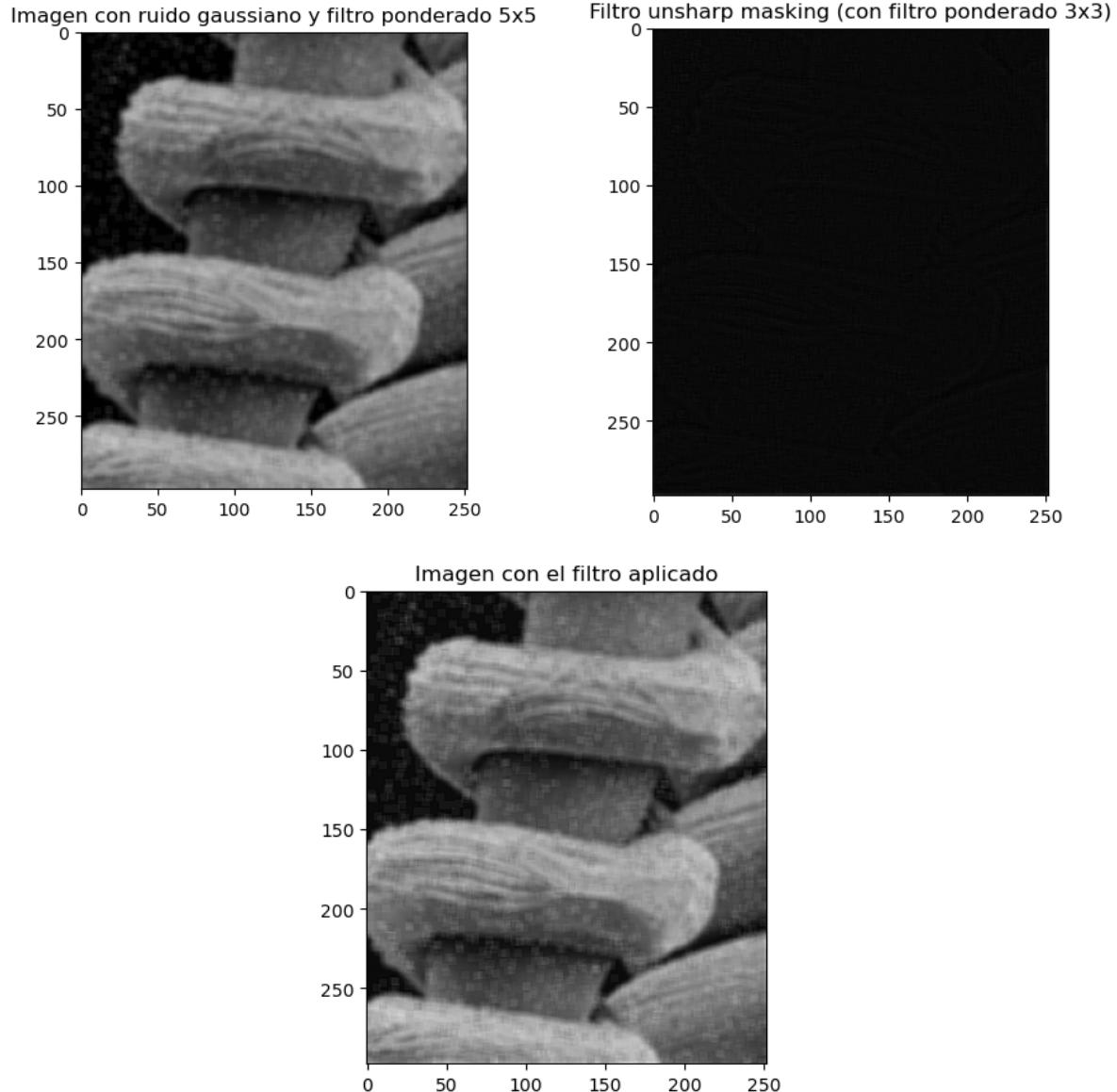
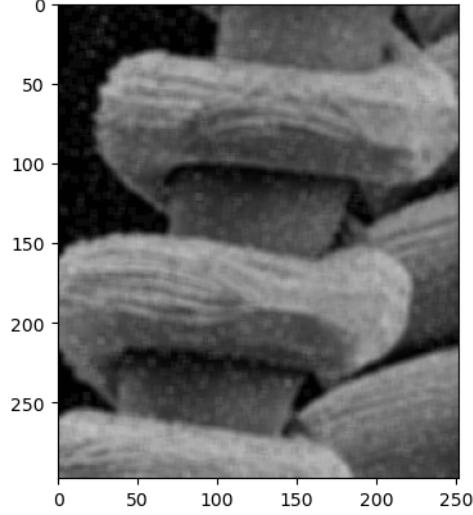


Imagen con ruido gaussiano y filtro ponderado 5x5



Filtro unsharp masking (con filtro ponderado 7x7)

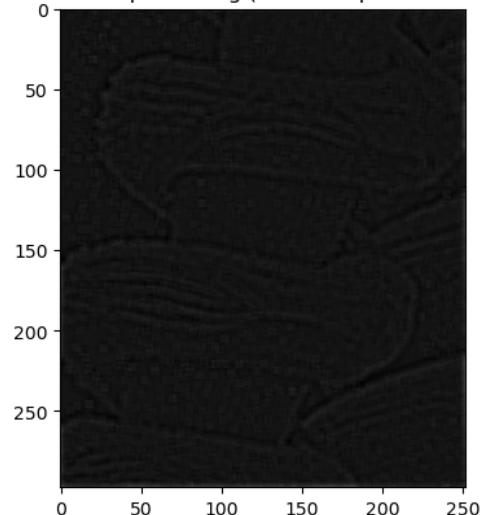
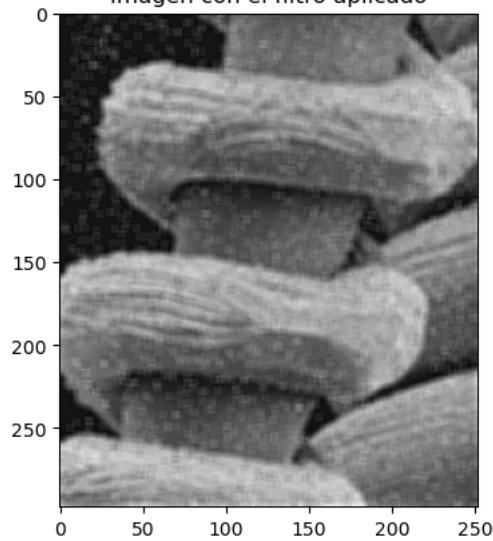


Imagen con el filtro aplicado



Conclusiones

Al realizar esta práctica se pudieron reforzar y comprender mejor la aplicación, funcionamiento y diferencias de los distintos procesamientos como transformaciones, aplicación de filtros, creación y análisis de los histogramas, que se pueden aplicar a las imágenes para mejorarlas espacialmente de acuerdo a nuestras necesidades.

En general, las transformaciones negativa, logarítmica y exponencial nos ayudan a ajustar el contraste o intensidades de una imagen. Por otro lado la ecualización del histograma nos permitió redistribuir las intensidades a lo largo del rango de intensidades de manera más uniforme y resultando una clara mejora de la imagen.

También los filtros vistos y aplicados, utilizaban operaciones de suavizado para intentar reducir el ruido; algunos mejor que otros y dependiendo del tipo de ruido, como lo fue el caso del filtro mediana manejando el ruido sal y pimienta. También tiene como factor importante el tamaño del kernel o máscara que se decida ocupar para cada caso.

Por otro lado los filtros basados en gradientes y laplacianos son buenos para la detección de bordes. También otros como sobel o prewitt permiten resaltar los bordes en diferentes direcciones. Y algunos de ellos como la aplicación del unsharp masking o el laplaciano nos pueden ayudar a acentuar estos bordes.

Referencias

1. Gonzalez, R., Woods, R., Digital Image Processing, Prentice Hall, 2008.