



# Tecnológico de Monterrey

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

*Análisis de Sistemas de Imagenología (Grupo 201)*

## Reporte Laboratorio 3. CT Reconstruction

### *Profesores*

Dr. José Gerardo Tamez Peña  
Dra. Diana Sofía Milagros Rosales Gurmendi

### *Equipo Pegaso:*

Daniela Alejandra Valdes Torres	A00832312
Diego De La Barreda Martínez	A01197739
Alexa María de León Durán	A01382990
Juan Luis Flores Sánchez	A01383088
Azul Sofía Moctezuma Enríquez	A01562585

*20 de abril de 2023*

## **Laboratorio 3. CT Reconstruction**

### **1. Introducción**

Los filtros de imágenes son herramientas importantes en el procesamiento de imágenes. En este caso se hablará sobre MTF, o función de transferencia de modulación, que es una medida de la capacidad de un sistema para transferir información sobre las frecuencias espaciales de una imagen. La MTF se utiliza a menudo para evaluar la calidad de los sistemas de imagen y los filtros de imagen. Los filtros de imagen tienen diferentes características de respuesta en frecuencia, y su MTF ayuda a determinar qué filtro es el más adecuado para una tarea específica de procesamiento de imágenes.

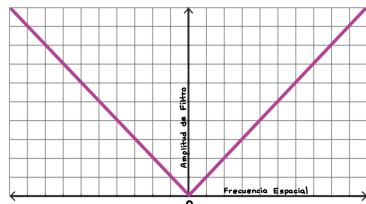
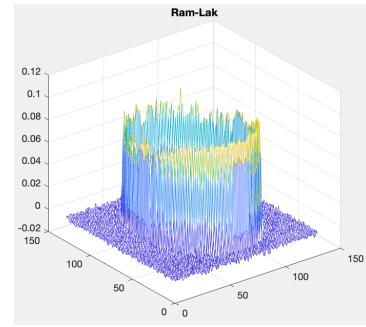
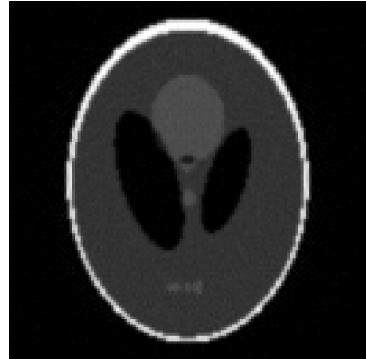
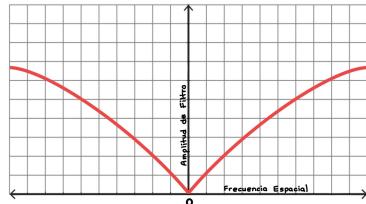
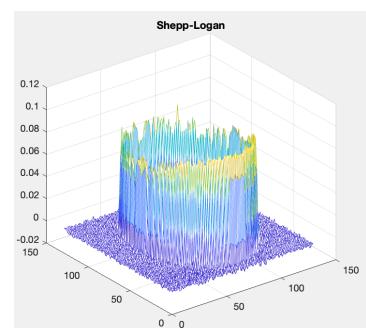
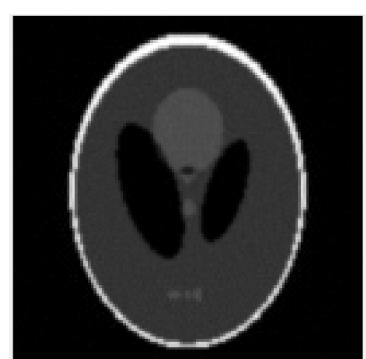
En el Anexo 2, se adjuntan los integrantes del equipo, estableciendo su participación porcentual, así como, el rol que desempeñaron.

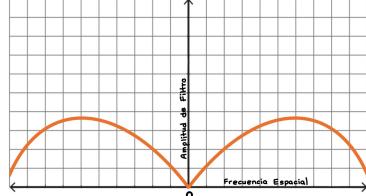
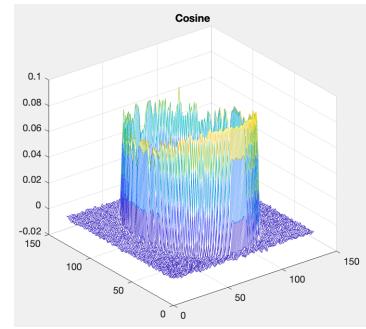
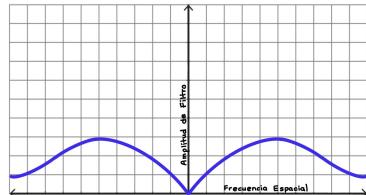
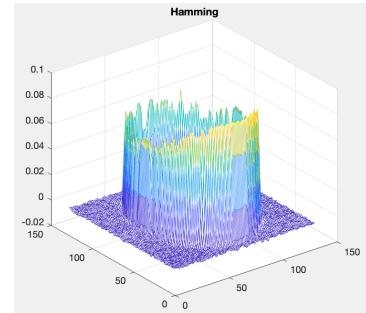
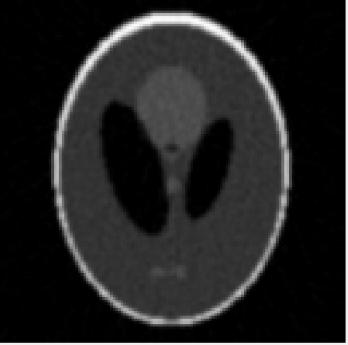
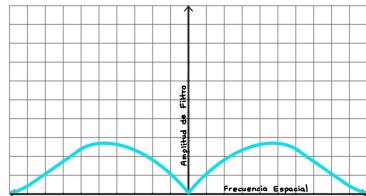
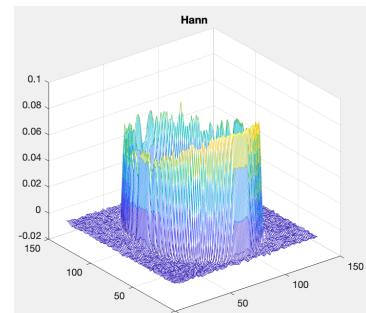
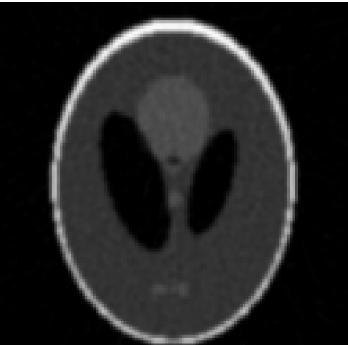
### **2. Objetivo**

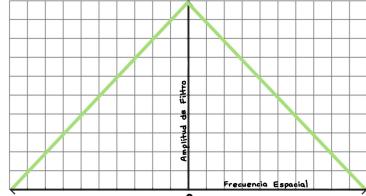
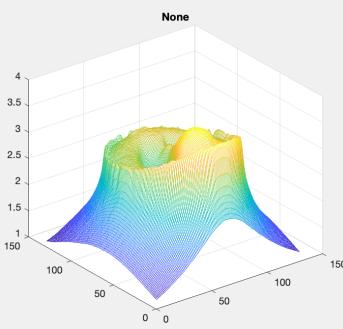
El objetivo de esta práctica es proporcionar una descripción clara de los diferentes filtros y agregar un gráfico de la función de transferencia de modulación (MTF) de cada ventana/filtro y de esta forma comprender las características de respuesta en la frecuencia de estos filtros.

### 3. Marco Teórico

En la Tabla 3.1 se adjuntan los tipos de filtros disponibles para la Transformada Inversa de Radon, su respectiva descripción, función de transferencia de modulación del sistema (*MTF*), su representación en 3D (*MESH*), y la imagen reconstruida con base en el filtro utilizado (*Filtered Back Projection*).

Tabla 3.1 Descripción y Resultados de Filtros en la Función de Transformada Inversa de Radon				
Filtro	Descripción	MTF	MESH	Filtered Back Projection
"Ram-Lak"	Filtro Ram-Lak o de rampa recortado. La respuesta en frecuencia de este filtro es $ f $ . Filtro sensible al ruido en las proyecciones.			
"Sheep-Logan"	Multiplica el filtro Ram-Lak por una función <b>sinc</b> .			

"Cosine"	Multiplica el filtro Ram-Lak por una función <b>cosine</b> .			
"Hamming"	Multiplica el filtro Ram-Lak por una <b>ventana Hamming</b> .			
"Hann"	Multiplica el filtro Ram-Lak por una <b>ventana Hann</b> .			

"None"	Sin filtro. <a href="#">iradon</a> devuelve los datos de retroproyección sin filtrar.	 <p>A graph showing a triangular filter function. The vertical axis is labeled "Amplitud del Filtro" and the horizontal axis is labeled "Frecuencia Espacial". The plot shows a triangle peaking at the center (0) with a maximum value of 1. The base of the triangle spans from -5 to 5 on the x-axis.</p>	 <p>A 3D surface plot titled "None" showing a smooth, bell-shaped surface. The vertical axis ranges from 0 to 4, and the horizontal axes range from 0 to 150. The surface is highest at the center and tapers off towards the edges.</p>	
--------	---	--	---	--

(MathWorks, 2023)

De acuerdo con la Figura 3.1, las principales diferencias que hay entre los filtros aplicados a la función iradon, en el filtro ‘Ram-Lak’ se puede observar que hace que la imagen sea mas nítida, el filtro ‘Shepp-Logan’ multiplica el filtro por la función sinc por lo que no existe un gran cambio, filtro ‘cosine’ minimiza la ISI y es utilizado para la transmisión de impulsos, el filtro ‘Hamming’ nos ayuda a utilizar el dominio del tiempo y la frecuencia provocando una restauración en las señales, el filtro ‘Hann’ separa y restaura las señales como el de ‘Hamming’ sin embargo, este filtro no cuenta con el escalón de la ventana y cuenta con una mayor amplitud, para finalizar el filtro ‘none’ en el que se puede ver que la función devuelve la proyección sin filtro. Los filtros Ram-Lak y Shepp-Logan son filtros de paso alto que mantienen intacta la información de los bordes, mientras que los filtros de coseno, hamming y hann son filtros de paso de banda, por lo que los primeros dos serían una buena opción para utilizar.

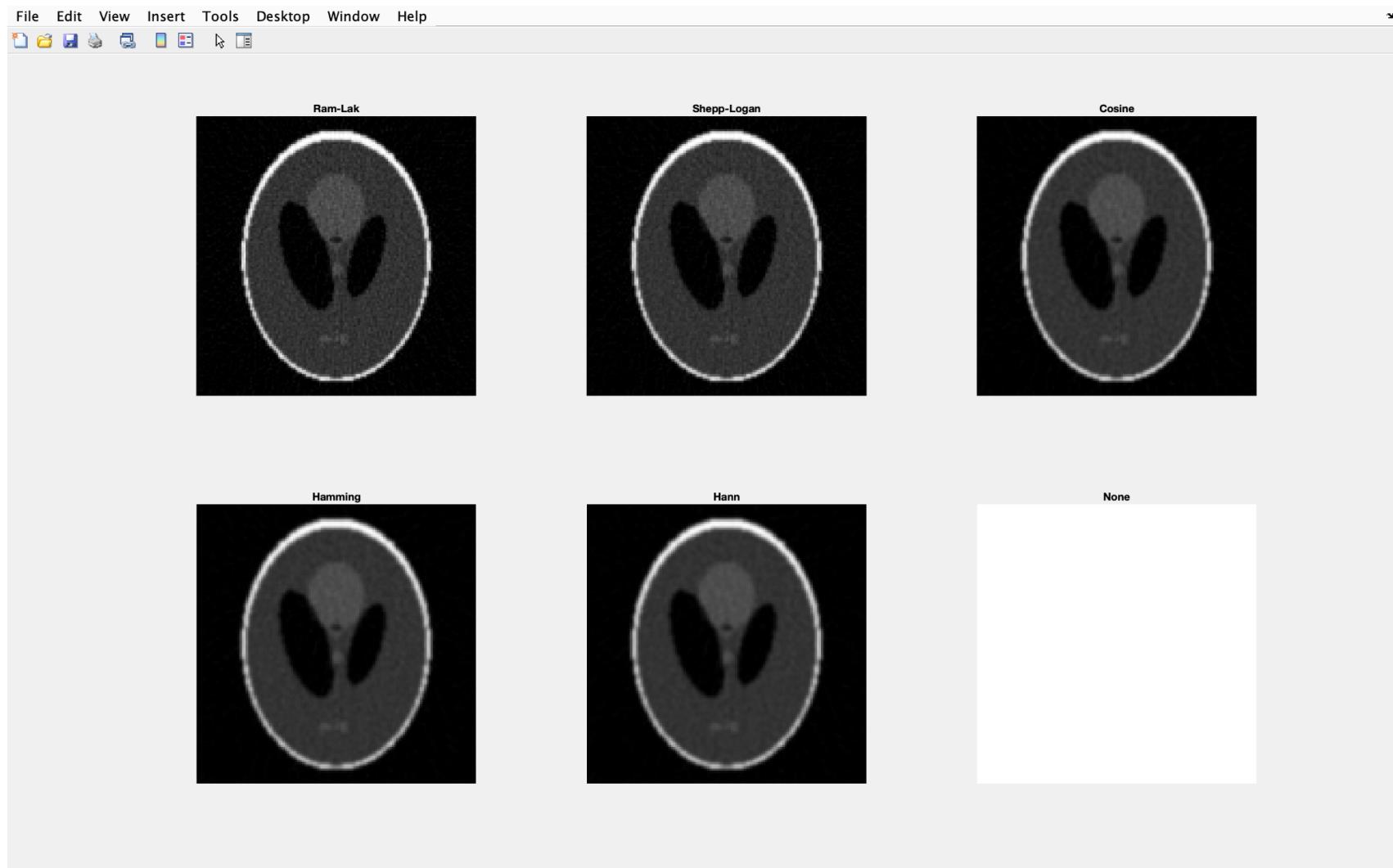


Figura 3.1. Imagen Reconstruida con la Función Inversa de Radón y los Diferentes Tipos de Filtros Disponibles

## **4. Metodología**

### **4.1 Materiales**

- MATLAB (MathWorks, Versión 2022b)
- Script (Véase Anexo 1)

### **4.2 Métodos**

#### **4.21 Descripción del Código**

En la reconstrucción de una imagen utilizando proyecciones con la transformada de Fourier en el programa de MATLAB, se definen los parámetros del planteamiento del problema, en este caso como primer parámetro se utiliza el número de ángulos de proyección, los fotones que llegan al detector, Tamaño de la imagen “sizeimage”, espaciado de fourier, ángulo de rotación y coeficientes de atenuación que posteriormente generarán una imagen sintética utilizando la función phantom de MATLAB, esta función se va modificando para así agregar puntos de alta densidad de atenuación (attgain).

Con la transformada de Fourier 2D se mostrará un subplot para generar la imagen sintética y reconstruirla con las proyecciones, la transformada de Radon realiza una inversión a la transformada reconstruyendo la imagen. Con la generación de ángulos de proyección para la transformada de Radon y aplicando su función en MATLAB para una corrección de atenuación incluyendo de igual forma ruido Poisson con los fotones transformando el resultado con una imagen de atenuación.

La función iradon nos indica una inversión de la transformada de Radon para generar tres versiones de la imagen construida: “Filtered Back Projection”, “cosine” (interpolación cúbica) y “none” (sin filtrado). Al realizar una reconstrucción de imágenes tomográficas con la transformada de Fourier y la transformada de Radon, en la transformada de Radon se debe aplicar a la imagen original “f” con la función “radon” para posteriormente incluir una corrección de atenuación y ruido con la función “exp”, “poissrnd” y “log”.

Después cuando las imágenes ya se encuentren corregidas con la transformada de Fourier, se interpola la imagen en el espacio de Fourier para tener una representación completa y así con la transformada inversa de Fourier se pueda obtener la imagen reconstruida con las debidas correcciones de fase, además se recorta la imagen resultante para eliminar los bordes agregados en la interpolación. El código puede mostrar visualmente las diversas etapas del proceso de reconstrucción en una cuadrícula de subtramas.

## 4.22 Parámetros del Simulador

Los parámetros generales para la simulación, se muestran en la Tabla 4.2:

Tabla 4.2 Parámetros Generales de la Simulación			
Tamaño de Imagen	128 píxeles por lado	Conteo de Fotones	1000 fotones
Espacio de Imagen	7 muestras por pixel	Coeficiente de Atenuación	0.25
Medio Círculo	180°		

En el experimento, se realizaron dos conjuntos de simulaciones de acuerdo con el valor de la variable *highdensityAttenuation* (Tabla 4.21); el primer valor es de 0.1 y corresponde a una toma de imágenes ideal, mientras que con el segundo valor, de 10, se realiza una reconstrucción considerando un objeto metálico, que atenúa más la señal que cualquier tejido en el fantoma creado, y que permite a algunos fotones “rebotar”.

Tabla 4.21 Valores de *highdensityAttenuation* por Conjunto de Simulaciones

Conjunto de Simulaciones #1	Conjunto de Simulaciones #2
0.1	10

Los parámetros utilizados en el primer conjunto de simulaciones, correspondiente a la variable *hidensityAttenuation* de 0.1, son aquellos que nos permiten identificar el cambio en la imagen reconstruida con diferentes tipos de filtros para la transformada inversa de Radon. El tipo de filtro para cada una de las simulaciones, se indica en la Tabla 4.22 y los resultados correspondientes se muestran en la Figura y Tabla 3.1:

Tabla 4.22 Conjunto de Simulaciones #1 - Tipo de Filtros para la Función *iradon()*

Filtro 1	Filtro 2	Filtro 3	Filtro 4	Filtro 5	Filtro 6
“Ram-Lak”	“Shepp-Logan”	“Cosine”	“Hamming”	“Hann”	“None”

Los parámetros utilizados en el segundo conjunto de simulaciones, correspondiente a la variable *highdensityAttenuation* de 10, son aquellos que nos permiten identificar el cambio en el procesamiento de la imagen con un diferente número de proyecciones de muestreo, los cuales se hacen de forma radial. El valor de este parámetro para cada simulación se indica en la Tabla 4.23:

Tabla 4.23 - Conjunto de Simulaciones #2 - Número de Proyecciones de Muestreo

Simulación 1	Simulación 2	Simulación 3	Simulación 4	Simulación 5
8	32	128	256	512

## 5. Resultados

### 5.1 Cambio de valor para el parámetro *highdensityAtenuation*



Figura 5.11 Resultados con *highdensityAtenuation* = 0.1.



Figura 5.12 Resultados con *highdensityAtenuation* = 10.

## 5.2 Reconstrucción de Imagen con Diferente Cantidad de Ángulos

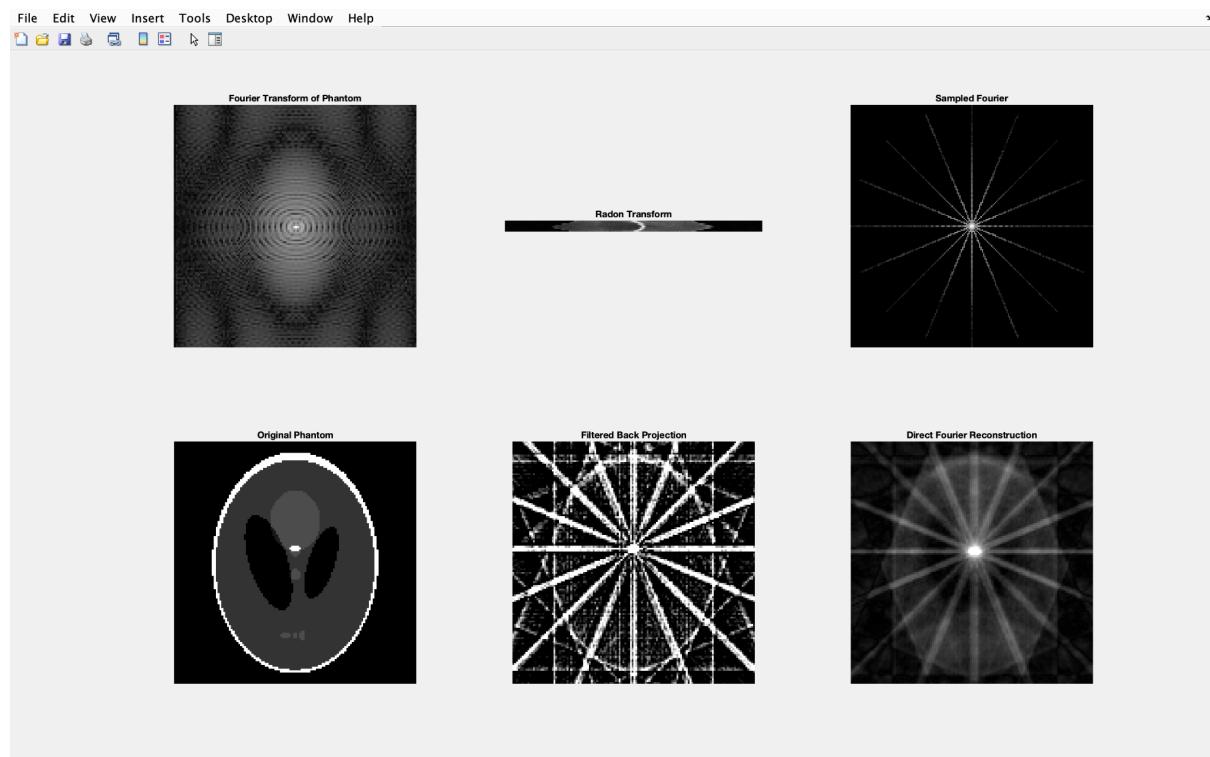


Figura 5.21 Resultados con 8 proyecciones muestreadas.

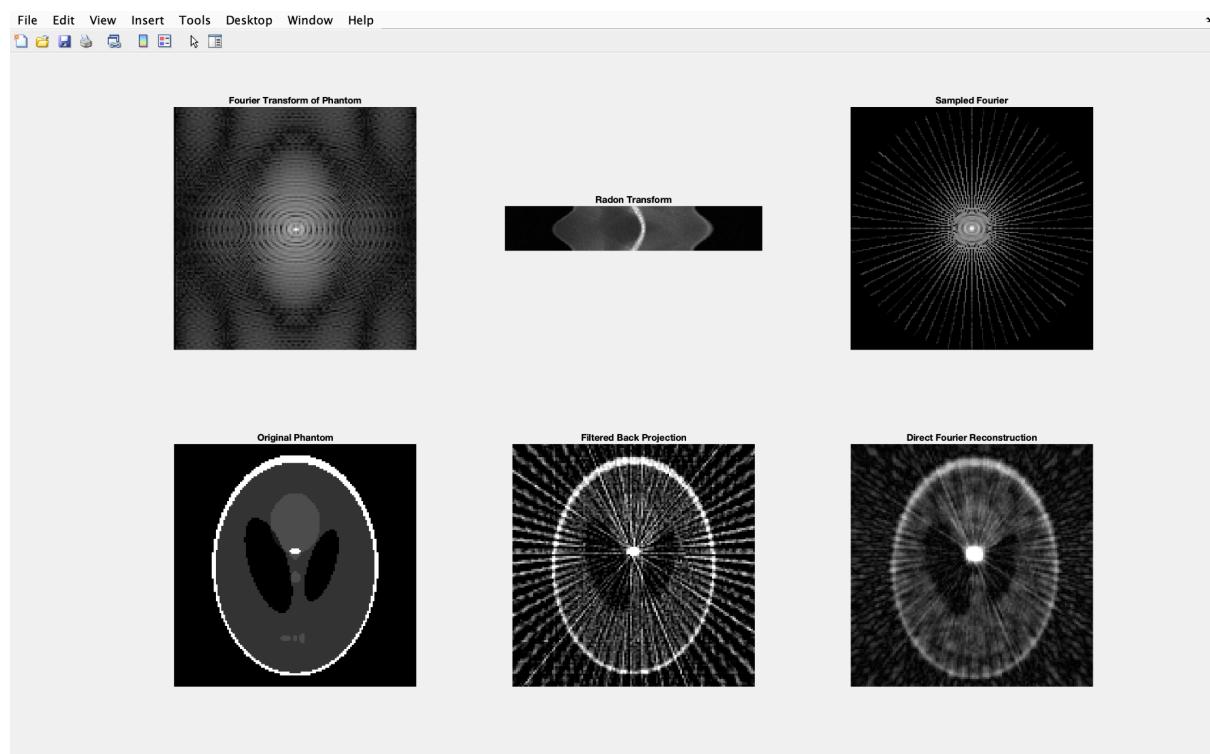
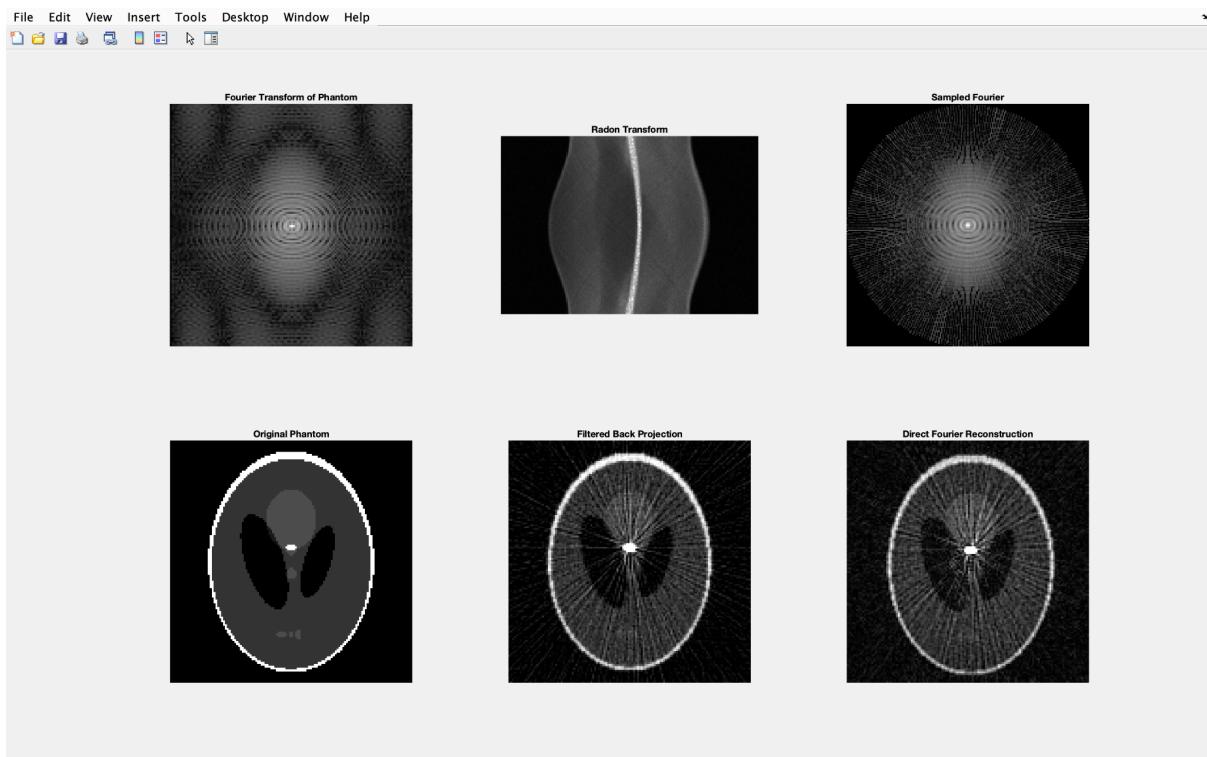
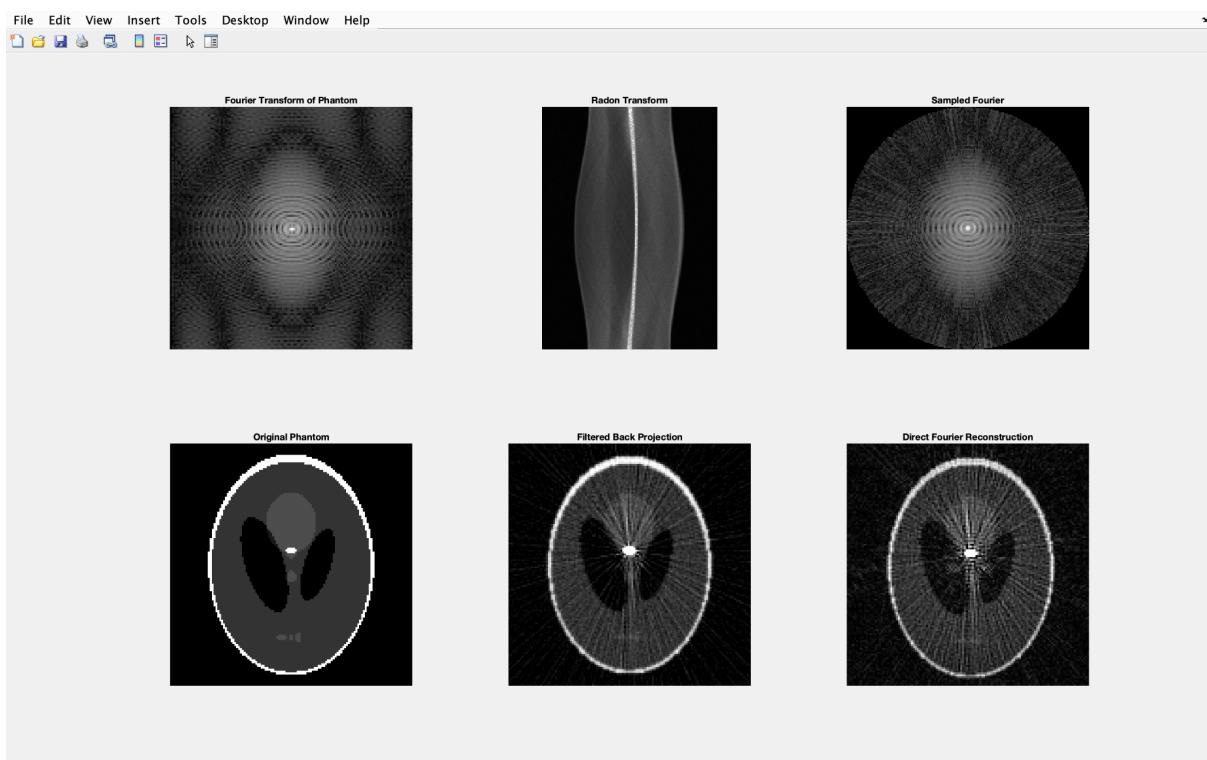


Figura 5.22 Resultados con 32 proyecciones muestreadas.



**Figura 5.23 Resultados con 128 proyecciones muestradas.**



**Figura 5.24 Resultados con 256 proyecciones muestradas.**

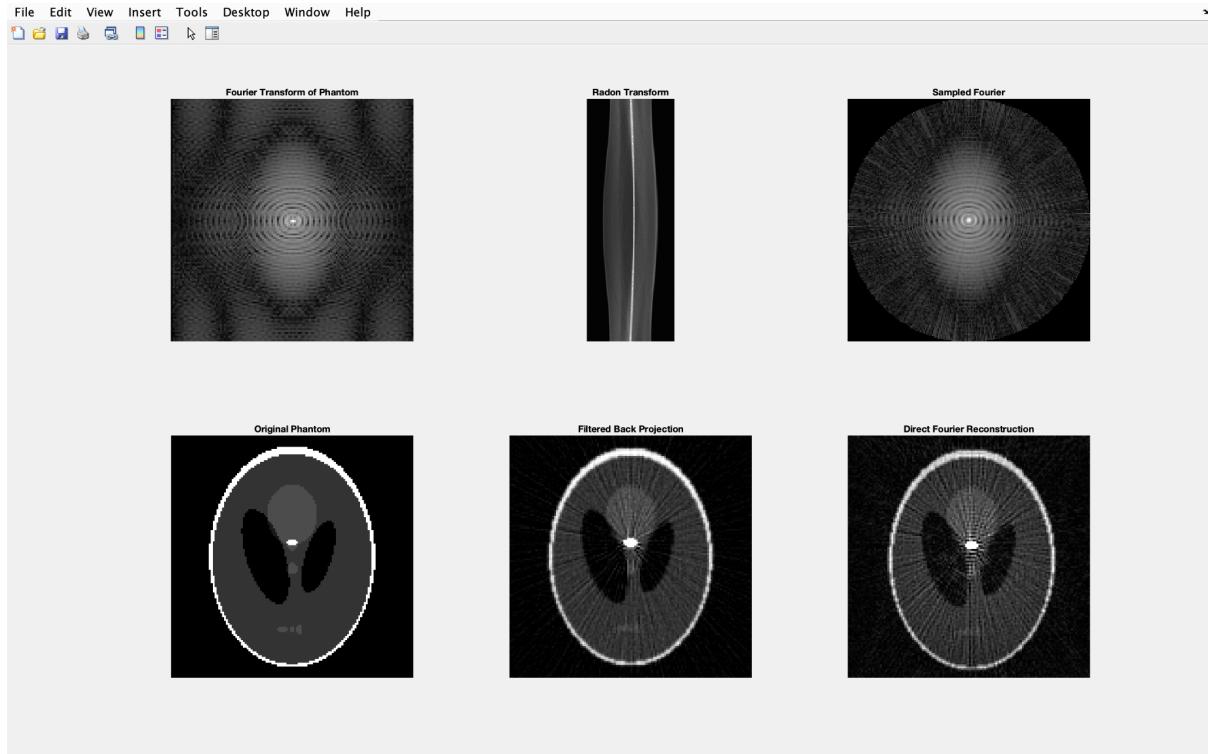


Figura 5.25 Resultados con 512 proyecciones muestreadas.

## 6. Discusión

Explica por qué las líneas del código comentadas (Líneas 78, 79, 80 y 114) no son correctas?

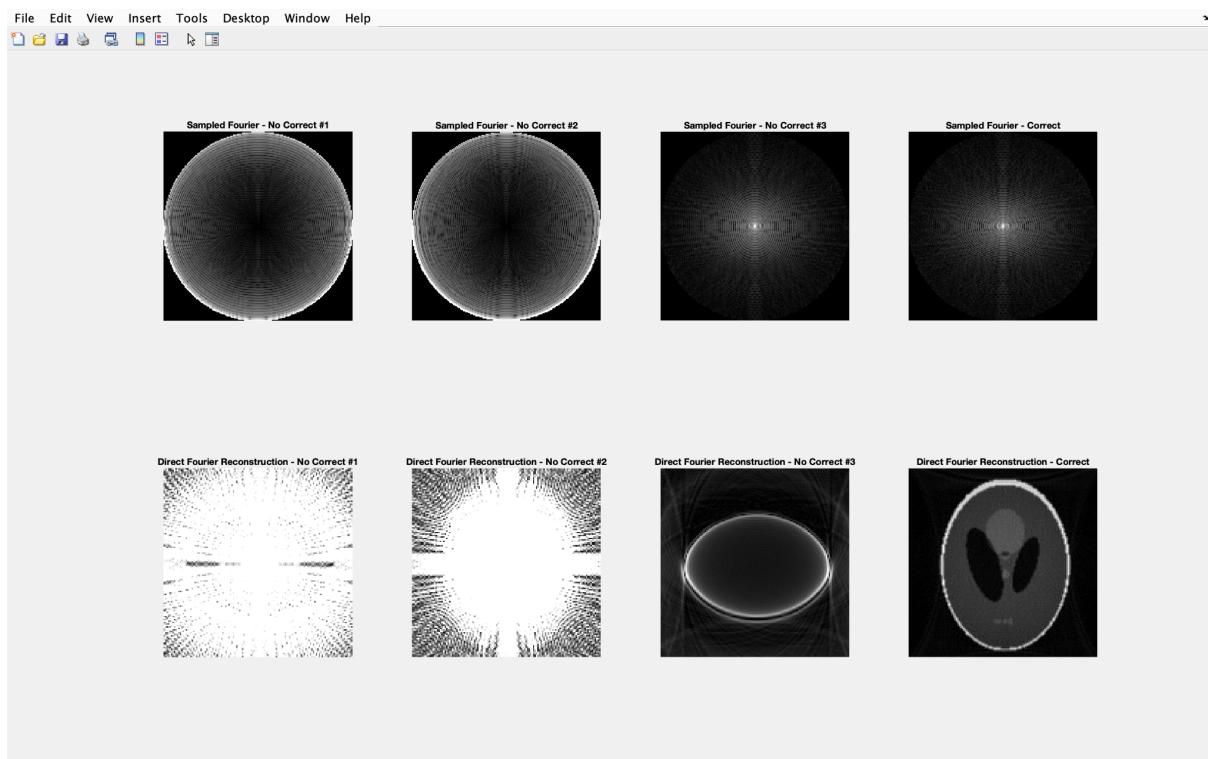
En cuanto a las expresiones mencionadas, se puede decir que g1, g2 y g3 (Tabla 6.1) son incorrectas porque no tienen en cuenta el desplazamiento (shift) que se produce en la señal al aplicar la FFT. En cambio, g4 es correcta ya que utiliza la función fftshift para corregir el desplazamiento en la señal y luego aplica la multiplicación por el factor shiftCorrection.

Tabla 6.1 Expresiones, Fórmula y Estado		
Expresión	Fórmula	Estado
g1	<code>g1 = fft(rad(:,n));</code>	Incorrecto
g2	<code>g2 = fft(rad(:,n)).*shiftCorrection;</code>	Incorrecto
g3	<code>g3 = fftshift(fft(rad(:,n)));</code>	Incorrecto
g4	<code>g4 = fftshift(fft(rad(:,n))). .* shiftCorrection;</code>	Correcto

Específicamente, se puede entender cada expresión de la siguiente manera:

- g1: Realiza la transformada de Fourier sobre la señal rad(:,n), pero no utiliza la función fftshift para corregir el desplazamiento, lo que puede afectar los resultados.

- g2: Realiza la transformada de Fourier sobre la señal  $\text{rad}(,:,n)$  y luego multiplica el resultado por el factor `shiftCorrection`, que puede no estar correctamente ajustado debido a la falta de corrección del desplazamiento en la señal.
- g3: Realiza la transformada de Fourier sobre la señal  $\text{rad}(,:,n)$  y utiliza la función `fftshift` para corregir el desplazamiento. Sin embargo, no se realiza la multiplicación por el factor `shiftCorrection`, lo que puede afectar la precisión de los resultados.
- g4: Realiza la transformada de Fourier sobre la señal  $\text{rad}(,:,n)$ , utiliza la función `fftshift` para corregir el desplazamiento y luego multiplica el resultado por el factor `shiftCorrection`, lo que debería proporcionar resultados más precisos y confiables.



**Figura 6.1 Comparación de Métodos Incorrectos y Correctos de Reconstrucción de Imágenes  
(De izq. a der. Incorrecto #1, Incorrecto #2, Incorrecto #3 y Correcto)**

### ¿Cuál es el propósito de la corrección de cambio (shift correction)? (Línea 75)

Cuando se utiliza la transformada de Fourier en MATLAB para analizar imágenes, la corrección de desplazamiento se utiliza para corregir el hecho de que el origen de la representación del dominio de la frecuencia de una imagen se encuentra en las esquinas de la imagen, en lugar de en su centro. Esto se debe a que el algoritmo FFT (Transformada rápida de Fourier) de MATLAB asume que la señal de entrada es periódica y, por lo tanto, envuelve la señal alrededor para crear una señal periódica sin interrupciones para su cálculo. Este envoltorio hace que el origen se encuentre en las esquinas de la imagen.

El propósito de la corrección de desplazamiento en este contexto es reposicionar el origen del dominio de la frecuencia en el centro de la imagen, de modo que las frecuencias se

alineen correctamente con las frecuencias espaciales de la imagen. Esto es importante para la correcta interpretación de la representación del dominio de la frecuencia de la imagen y para realizar más operaciones de procesamiento de imágenes basadas en esta representación.

### **¿Cuál es el propósito de la corrección de fase (phase correction)? (Línea 111 y 112)**

Cuando se utiliza la transformada de Fourier en MATLAB para analizar imágenes, la corrección de fase se utiliza para corregir cualquier desplazamiento de fase que pueda ocurrir en el sistema. El desplazamiento de fase puede ser causado por una variedad de factores, como variaciones en las condiciones de iluminación, ruido en el proceso de adquisición de imágenes o imperfecciones en la lente de la cámara. Estos factores pueden introducir errores de fase en los datos de la imagen, lo que puede afectar la precisión de la transformada de Fourier.

El propósito de la corrección de fase en este contexto es eliminar o minimizar estos errores de fase y garantizar que la transformada de Fourier represente con precisión los datos de imagen subyacentes. Esto es importante para realizar operaciones de procesamiento de imágenes precisas basadas en la transformada de Fourier, como filtrado o compresión.

### **Discute las diferencias y similitudes de la conducta de la Reconstrucción Directa de Fourier y la Transformada Inversa de Radon por Pasos**

La Reconstrucción Directa de Fourier (FD) y la Transformada Inversa de Radon por Pasos (SIRT) son dos técnicas utilizadas para la reconstrucción de imágenes a partir de proyecciones o medidas de diferentes tipos.

La FD se basa en la Transformada de Fourier y utiliza la información de las proyecciones en el dominio de Fourier para reconstruir la imagen original. La FD es un método rápido y eficiente para la reconstrucción de imágenes, pero puede ser susceptible a artefactos debido a la falta de información de fase. Por otro lado, la SIRT es un método iterativo que utiliza la información de las proyecciones en el dominio de la imagen para reconstruir la imagen original. La SIRT es más robusta que la FD y puede manejar datos incompletos o ruidosos. Sin embargo, la SIRT es un proceso iterativo que puede ser computacionalmente costoso. En cuanto a las similitudes, ambas técnicas se utilizan para la reconstrucción de imágenes a partir de datos de proyección y son ampliamente utilizadas en tomografía computarizada, resonancia magnética y otras técnicas de imagen médica.

Por lo tanto, la FD y la SIRT son dos técnicas diferentes para la reconstrucción de imágenes a partir de datos de proyección. Mientras que la FD es rápida y eficiente, pero puede ser susceptible a artefactos, la SIRT es más robusta y puede manejar datos incompletos o ruidosos, pero puede ser computacionalmente costosa debido a su naturaleza iterativa.

## **7. Conclusiones**

**Azul Sofía Moctezuma Enriquez:** Las técnicas de reconstrucción son una herramienta muy completa y eficiente para realizar tomografías computarizadas, es de suma importancia que se consideren diversos parámetros como los filtros y ángulos de proyección para que los resultados sean precisos, con la transformada de Radon y transformada de Fourier se pudieron poner en práctica diversas funciones con la herramienta de MATLAB que hicieron posible la generación de imágenes útiles para fines médicos, además de lograr identificar los filtros aplicados a las imágenes y cómo estas pueden mejorar o reducir su calidad.

**Diego Alejandro De la Barreda Martínez:** Con esta práctica se pudo reforzar los conceptos de transformación de imágenes así como correcciones, proyecciones y reconstrucción de esta misma. Se aplican diferentes técnicas para la imagen y una herramienta (MFT) para poder calcular la relación que existe entre el contraste y poder de resolución, señalando la nitidez y resolución de la imagen. En los resultados pudimos apreciar cómo se reconstruyen las imágenes de acuerdo a diferentes ángulos y atenuaciones. Esto aplicable en los campos de imagenología con más importancia en la sección de CT al reconstruir imágenes médicas obtenidas a través de este estudio.

**Alexa María de León Durán:** Durante esta práctica se realizó una reconstrucción de una imagen de Tomografía Computarizada en donde pudimos trabajar con MTF que es una herramienta fundamental para poder evaluar la calidad de los filtros de imagen, ya que nos muestra cómo cada filtro afecta las frecuencias espaciales en una imagen, además se aplicaron la transformada de Radon que reconstruye imágenes a partir de proyecciones y la transformada de Fourier. Una vez que conocemos y analizamos las características de respuesta en frecuencia de los filtros de imagen, se puede identificar de una forma más sencilla el tipo de filtro más adecuado para una tarea específica de procesamiento de imágenes.

**Juan Luis Flores Sánchez:** Con base en la práctica realizada, fue posible reforzar y poner en práctica los conocimientos teóricos y prácticos de sistemas de obtención y transformación de imágenes, relacionándolos con tópicos como los fundamentos físicos, matemáticos y de análisis de señal para la obtención de imágenes de Tomografía Computarizada (CT), transformada de Fourier, transformada de Radon, función de transferencia de modulación del sistema (MTF), así como algunos aspectos como coeficientes de atenuación y la conducta de los filtros en la construcción de imágenes. De esta manera, utilizando un fantoma artificial (desarrollado en computadora, Fantoma de Radón) que nos permite simular diferentes tejidos, fue posible comprender y dimensionar la reconstrucción de una imagen tomada por Tomografía Computarizada y el procesamiento de las mismas, inclusive ante situaciones como un objeto metálico que se encuentra en la cabeza, muestreo con diferente número de proyecciones, y los filtros que pueden ser utilizados al momento de identificar enfermedades. Para ello, fue necesario la utilización de herramientas computacionales como MATLAB, así como un claro entendimiento de sus comandos, funciones y el orden de aplicación de las mismas.

**Daniela Alejandra Valdes Torres:** En la práctica realizada se pudo realizar la reconstrucción de la imagen de CT con la transformada de Fourier y las funciones de MATLAB, además de la transformada de Radon para crear una proyección en la imagen y reconstruirla a través de proyecciones, se implementaron diversos métodos para mejorar la calidad de la imagen con funciones como la corrección de fase, aplicación de pesos a las proyecciones y la eliminación de bordes. Esta reconstrucción de imagen por medio de la transformada de Fourier es una de las técnicas más utilizadas en las tomografías computarizadas (CT) y otras aplicaciones proporcionando herramientas útiles con la implementación de este método y optimización de resultados.

## 8. Referencias

- [1]. MathWorks. (2023). Transformada de Radon Inversa. Abril 20, 2023. Recuperado de: <https://la.mathworks.com/help/images/ref/iradon.html#d124e57472>

## 9. Anexos

### Anexo 1. Script de MATLAB (“ctreconstructionsimple.m”)

```
%% BI2007B Análisis de Sistemas de Imagenología
%% Equipo PEGASO

%% Reconstruction from projections
%%%%%%%%%%%%% Parameters
NumAngles = 512; % Number of sampled projections
sizeimage = 256; %Image size
fsSpacign = 7; %Fourier Transform samples per pixel
halfcircle = 180; % Rotation degrees
photonCount = 1.0e4; % Photons that reached the detector
attgain = 0.25; % to change the attenuaiton coefficient
highdensityAtenuation = 0.1;
% Synthetic image (Phantom plus dots)
attgain = 50 * attgain/sizeimage;
f=abs(phantom(sizeimage));
% imshow(f,[])
f((f > 0.35) & (f < 0.45)) = highdensityAtenuation;
f = attgain*f;
%f(1:sizeimage/2,1:sizeimage/2) = phantom(sizeimage/2);
%f(sizeimage/2+1:sizeimage,sizeimage/2+1:sizeimage) =
phantom(sizeimage/2)';
%f(sizeimage/2,sizeimage/2) = 1;
%f(round(2*sizeimage/3),round(1*sizeimage/3)) = 1;
%f(round(1*sizeimage/3),round(2*sizeimage/3)) = 1;
%figure(1)
%imshow(f,[])
figure(2)
subplot(2,3,4)
```

```
imshow(f,[0,attgain])
title('Original Phantom')
FF = fft2(f);
subplot(2,3,1)
imshow(log(abs(fftshift(FF))+1),[])
title('Fourier Transform of Phantom')

%% Radon
% The projection angles and the radon transfrom
angles =
linspace(0,halfcircle*(1.0-1.0/NumAngles),NumAngles);
rad = radon(f,angles);
rad = exp(-rad);
rad = double(uint16(poissrnd(photonCount*rad)));
rad(rad<1) = 0.5;
rad = -log(rad/photonCount);
subplot(2,3,2)
imshow(rad',[])
title('Radon Transform')

%% Inverse Radon Transform
rec = iradon(rad,angles,'linear');
rec1 = iradon(rad,angles,'linear','cosine');
rec2 = iradon(rad,angles,'linear','none');
subplot(2,3,5)
imshow(rec,[0,attgain])
title('Filtered Back Projection')

% figure(3)
% subplot(231)
% imshow(rec,[0,attgain])
% subplot(232)
% imshow(rec1,[0,attgain])
% subplot(233)
```

```

% imshow(rec2,[0,attgain])
% subplot(234)
% mesh(rec)
% subplot(235)
% mesh(rec1)
% subplot(236)
% mesh(rec2)

%% Direct Fourier Reconstruction
sz0 = size(f);
srad = size(rad);
fff = complex(zeros(srad(1)));
sz = size(fff);
wts = zeros(sz(1));
shiftCorrection = exp(pi()*j*(srad(1)+1.0)/srad(1)*(1:srad(1)))';
n=1;
for i=angles
    g = fft(rad(:,n)); %not
    correct
    g = fft(rad(:,n)).*shiftCorrection; %not
    correct
    g = fftshift(fft(rad(:,n))); %not
    correct
    g = fftshift(fft(rad(:,n))).*shiftCorrection; %the
    correct way
    sg = size(g);
    theta = i*pi()/180.0;
    sintheta = sin(theta);
    costheta = cos(theta);
    halfline = sg(1)/2.0 + 1.0;
    halfimage = round(sz(1)/2.0);
    sratio = sz(1)/sg(1);
    for an=0:(fsSpacign*sg(1)+1)
        fidx = (an+0.5)/fsSpacign;
        r = sratio*(fidx - halfline);
        xr = r*costheta;
        yr = r*sintheta;
        col = floor(xr+0.5) + halfimage;
        row = halfimage - floor(yr+0.5);
        gidx = floor(fidx);
        if ((row <= sz(1)) && (row > 0) && (col <= sz(1)) && (col > 0) && (gidx > 0) && (gidx <= sg(1)))
            w1 =
            exp(-((round(xr)-xr)^2+(round(yr)-yr)^2)/0.5);
            fff(row,col) = fff(row,col) + w1*g(gidx);
            wts(row,col) = wts(row,col) + w1;
        end
    end
    n = n + 1;
end
fff(wts>0) = fff(wts>0)./wts(wts>0);
subplot(2,3,3)
imshow(log(abs(fff)+1),[])
title('Sampled Fourier')
phaseCorrection = exp(-pi()*li*(1:sz(1))');
phaseCorrection = phaseCorrection.*phaseCorrection';
%frec = abs((ifft2(fff))); %This is not correct
%frec = abs((ifft2(fff))); %Not correct
frec = abs((ifft2(fff .* phaseCorrection))); %The
correct way
deltacrop = round((sz(1)-sz0(1))/2);
frec =
frec(deltacrop:(sz(1)-deltacrop),deltacrop:(sz(1)-delta
crop));
subplot(2,3,6)
imshow(frec,[0,attgain]);
title('Direct Fourier Reconstruction')

```

**Anexo 2. Tabla de Integrantes, Participación y Roles en el Laboratorio**

<i>Tabla 1. Integrantes, Participación y Roles en el Laboratorio</i>		
<i>Integrante</i>	<i>Participación</i>	<i>Rol en la Práctica</i>
Alejandra Valdes	20%	Metodología
Diego de la Barreda	20%	Marco Teórico
Alexa de León	20%	Introducción, Objetivo
Juan Luis Flores	20%	Resultados
Sofía Moctezuma	20%	Discusión