

# PROCESAMIENTO DIGITAL DE SEÑALES

ALUMNO: POSE, FERNANDO EZEQUIEL

LEGAJO: 143.791-4

CICLO LECTIVO: 2° CUATRIMESTRE 2015

PROFESOR: DR. ING. MARIANO LLAMEDO SORIA

## TRABAJO PRÁCTICO 5 - EJERCICIO 1

Implementar el algoritmo que calcule la ecuación de diferencias para la implementación direct form 1() de filtros IIR.

$$Y(z) = X(z) \cdot (b_0 + b_1 z^{-1} + \dots + b_N z^{-N}) - Y(z) \cdot (a_1 z^{-1} + \dots + a_M z^{-M})$$

Compruebe su funcionamiento con algunos filtros diseñados mediante la herramienta fdatool de Matlab.

Se implementó el algoritmo requerido por la consigna del trabajo práctico. Directform1() cuya función en Matlab se denomina filter(). Dicho código puede observarse al final de este ejercicio. Además el mismo se adjunta al trabajo práctico en la carpeta “funciones básicas” con el nombre de: “MyDirectFrom”.

Para la comprobación del correcto funcionamiento de la función implementada se generó la señal tritonal cuyos parámetros se mencionan a continuación.

Señal de prueba:

Amplitud	Offset	Muestras	Fs	Phase
1V	0	100	500Hz	0

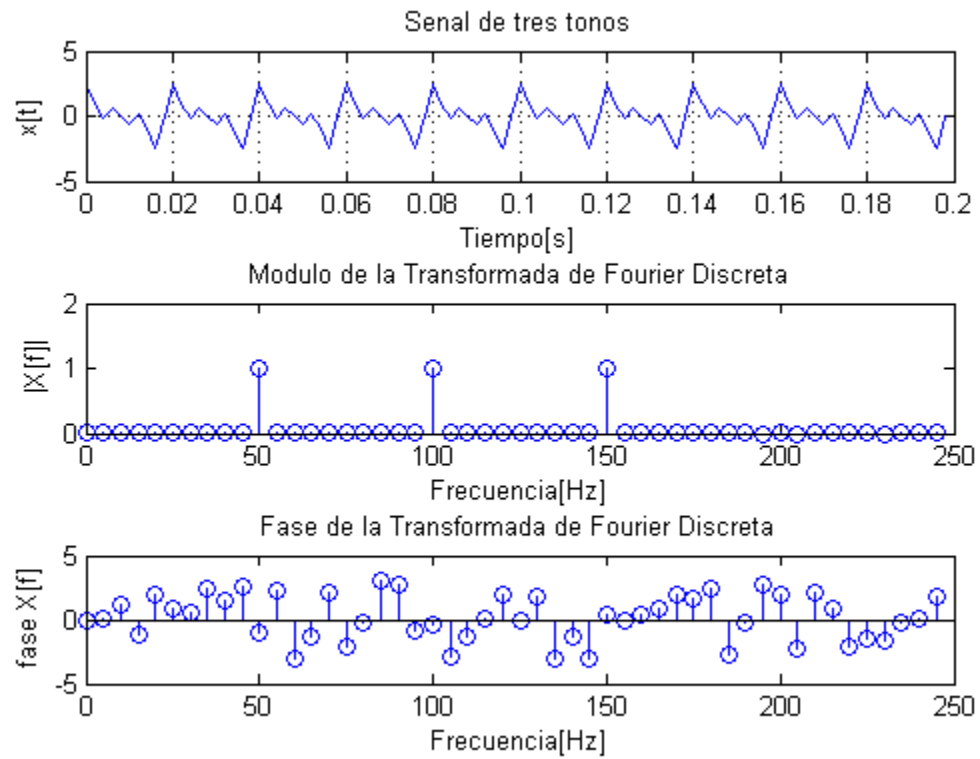
Fo1	Fo2	Fo3
50Hz	100Hz	150Hz

```
signal_1 = fsenoidal(amp, fo_1, phase, offset, N, fs);  
signal_2 = fsenoidal(amp, fo_2, phase, offset, N, fs);  
signal_3 = fsenoidal(amp, fo_3, phase, offset, N, fs);
```

% Obtengo la señal con la que voy a trabajar:

```
signal = signal_1 + signal_2 + signal_3;
```

Señal tritonal generada – Tiempo y Frecuencia



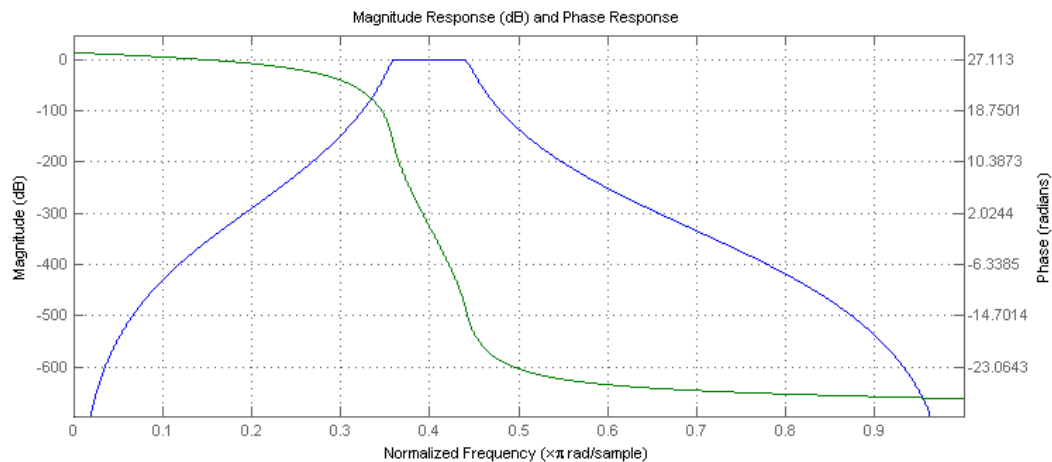
Para verificar el correcto funcionamiento de la función se le aplico un filtro pasabanda a dicha señal cuyos parámetros y diseño se expone a continuación:

```
Fs = 500; % Sampling Frequency

Fstop1 = 80;      % First Stopband Frequency
Fpass1 = 90;      % First Passband Frequency
Fpass2 = 110;     % Second Passband Frequency
Fstop2 = 120;     % Second Stopband Frequency
Astop1 = 100;     % First Stopband Attenuation (dB)
Apass  = 1;       % Passband Ripple (dB)
Astop2 = 30;      % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                   Astop2, Fs);
Hd = design(h, 'butter', 'MatchExactly', match);
```

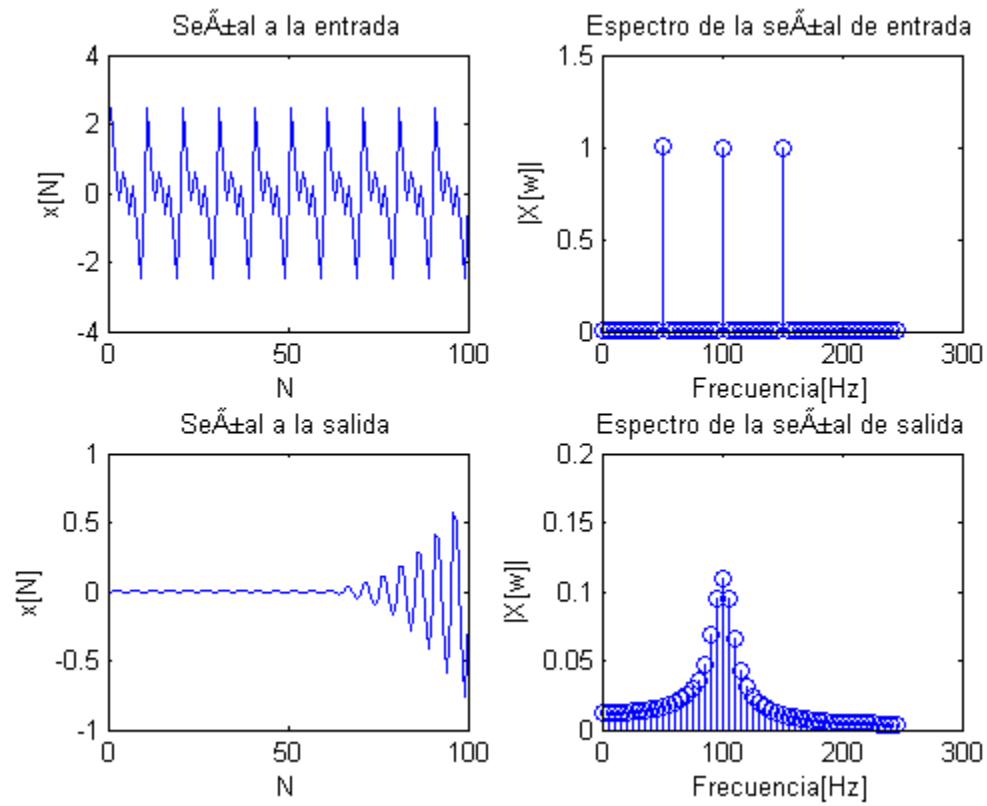
Respuesta en frecuencia del filtro diseñado



En la imagen se puede observar la respuesta en frecuencia del filtro cuyas referencias de las curvas expuestas son:

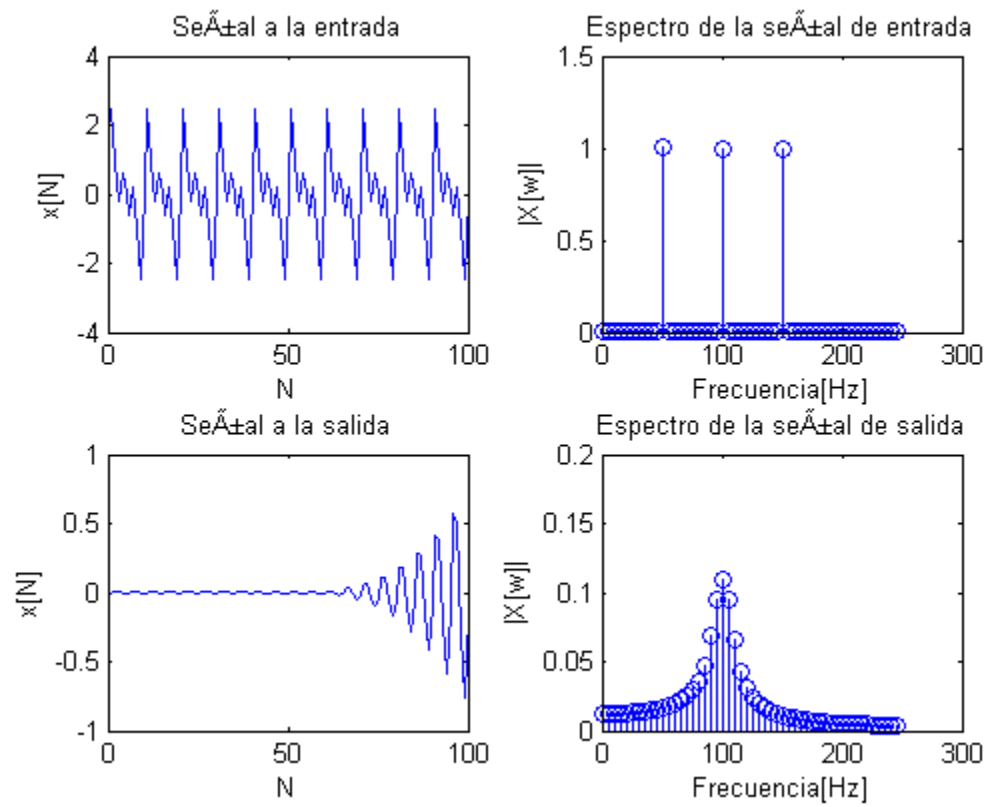
- En azul se puede observar el módulo de la respuesta en frecuencia del filtro.
- En verde se puede observar la fase de la respuesta en frecuencia del filtro.

Al aplicar el filtro pasabanda a la señal tritonal generada utilizando la función `filter()` se obtuvieron los siguientes resultados:



En los gráficos se puede observar la señal de entrada, el espectro de la señal de entrada y luego la señal a la salida del filtro y su correspondiente espectro.

Luego se volvió a realizar el mismo experimento, pero en este caso se utilizó el algoritmo generado en reemplazo a la función `filter()` obteniéndose los siguientes resultados:



Finalmente se puede observar que la señal obtenida a la salida del filtro coincide con la señal obtenida en el caso en el cual se utilizaba las funciones de Matlab por lo que se puede concluir en que el algoritmo funciona de forma correcta.

## Ejercicio 1 – Códigos

A continuación se exponen los principales códigos generados para la resolución del ejercicio

### Ejercicio 3 - Código: MyDirectFrom1

```
function [out] = MyDirectFrom(X,B,A)

    LengthB = length(B);
    LengthX = length(X);

    if (isrow(X) == 0)
        X = X';
    end

    XX = [zeros(1,LengthB-1),X];
    YY = [zeros(1,LengthB-1),zeros(1,LengthX)];

    A = 1/A(1) * A;
    B = 1/A(1) * B;

    Bflip = fliplr(B);
    Aflip = fliplr(A);

    for k = LengthB:LengthB+LengthX-1
        Sum1 = Bflip(1:end) * XX(k-(LengthB-1):k)';
        Sum2 = Aflip(1:end-1) * YY(k-(LengthB-1):k-1)';
        YY(k) = Sum1-Sum2;
    end
    out = YY(LengthB:LengthX+LengthB-1)';

end
```



## TRABAJO PRÁCTICO 5 - EJERCICIO 2

Mediante fdatool diseñe un filtro pasabanda con las siguientes características:

```
Fstop1 = 0.01;      % First Stopband Frequency
Fpass1 = 0.1;       % First Passband Frequency
Fpass2 = 35;        % Second Passband Frequency
Fstop2 = 50;        % Second Stopband Frequency
Astop1 = 60;        % First Stopband Attenuation (dB)
Apass  = 1;         % Passband Ripple (dB)
Astop2 = 80;        % Second Stopband Attenuation (dB)
```

Mediante las aproximaciones de Butterworth, Chebyshev 1 y 2 y elíptica. Aplique dichos filtros a la señal ECG1 provista en el archivo TP4\_ECG.mat que acompaña esta práctica. Este tipo de filtrado forma parte del procesamiento típico para cualquier análisis automático de la señal electrocardiográfica. Conteste las siguientes preguntas:

- 1) ¿La señal filtrada se encuentra en fase con la señal original? ¿Cómo podría compensar dicho desfase? Genere una solución adecuada y presente los resultados.
- 2) Dado que las ondas presentes en el ECG contienen información acerca de la actividad eléctrica del corazón, ¿Podría asegurar que este procesamiento afecta a la morfología de las ondas de la manera esperada? Si afectara la morfología de manera inesperada explique el fenómeno y proponga una forma de mitigarlo.
- 3) Una vez que el filtro se comporta de una manera adecuada sin introducir distorsiones, modifique las frecuencias de corte superior e inferior hasta comenzar a afectar la morfología de las ondas PQRST. En base a esta experimentación estime aproximadamente las componentes espectrales de dichas ondas.
- 4) ¿Podrían filtrarse los movimientos bruscos de línea de base que ocurren por ejemplo en torno a la muestra 1e5, 4.4e5 o 6e5? Implemente una solución adecuada.
- 5) Repita los puntos anteriores para las otras señales de ECG.

Mediante la herramienta fdatool se diseñaron los filtros pasabanda Butterworth, Chebyshev 1 y 2 y elíptica con los parámetros indicados en el trabajo práctico. A continuación se expone el código generador de dichos filtros.

```
Fstop1 = 0.01;      % First Stopband Frequency
Fpass1 = 0.1;       % First Passband Frequency
Fpass2 = 35;        % Second Passband Frequency
Fstop2 = 50;        % Second Stopband Frequency
Astop1 = 60;        % First Stopband Attenuation (dB)
Apass  = 1;         % Passband Ripple (dB)
Astop2 = 80;        % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly
Fs     = 1000;      % Sampling Frequency

h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                   Astop2, Fs);

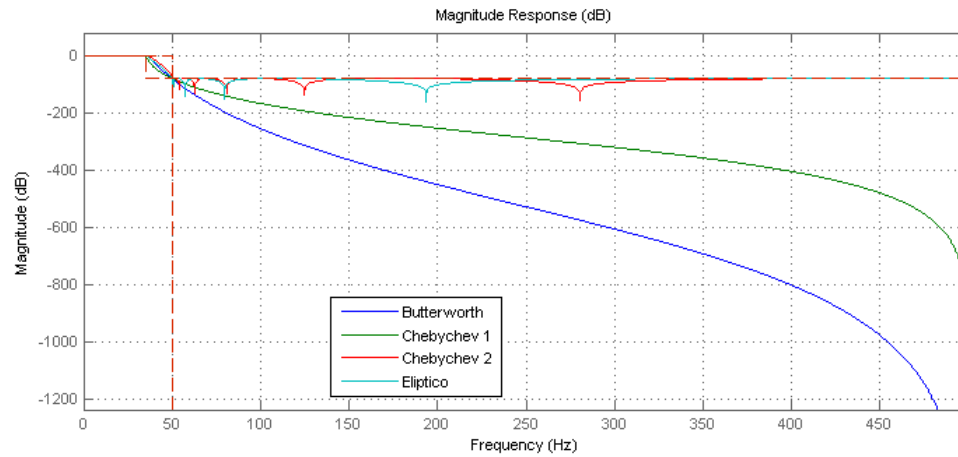
% Butterworth Bandpass filter
Butterworth = design(h, 'butter', 'MatchExactly', match);

% Chebyshev Type I Bandpass filter
ChebyshevI = design(h, 'cheby1', 'MatchExactly', match);

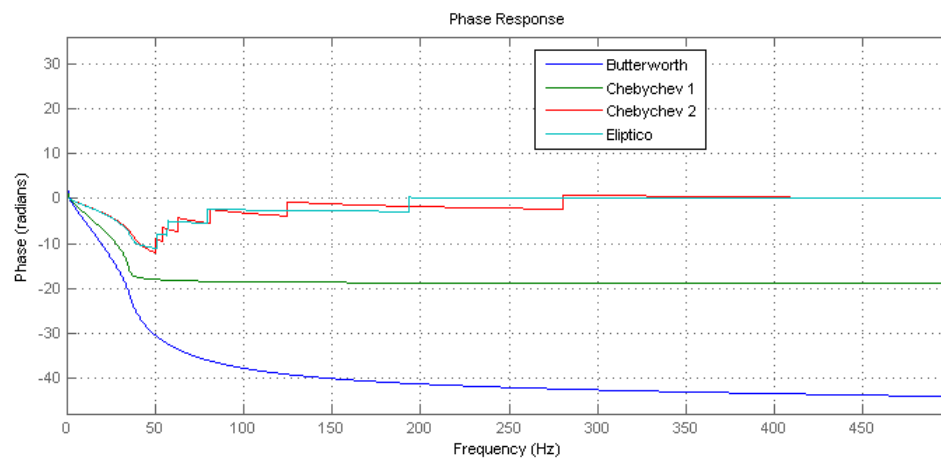
% Chebyshev Type II Bandpass filter
ChebyshevII = design(h, 'cheby2', 'MatchExactly', match);

% Elliptic Bandpass filter
Elliptic = design(h, 'ellip', 'MatchExactly', match);
```

## Respuesta en frecuencia de los filtros diseñados - Módulo



## Respuesta en frecuencia de los filtros diseñados – Fase

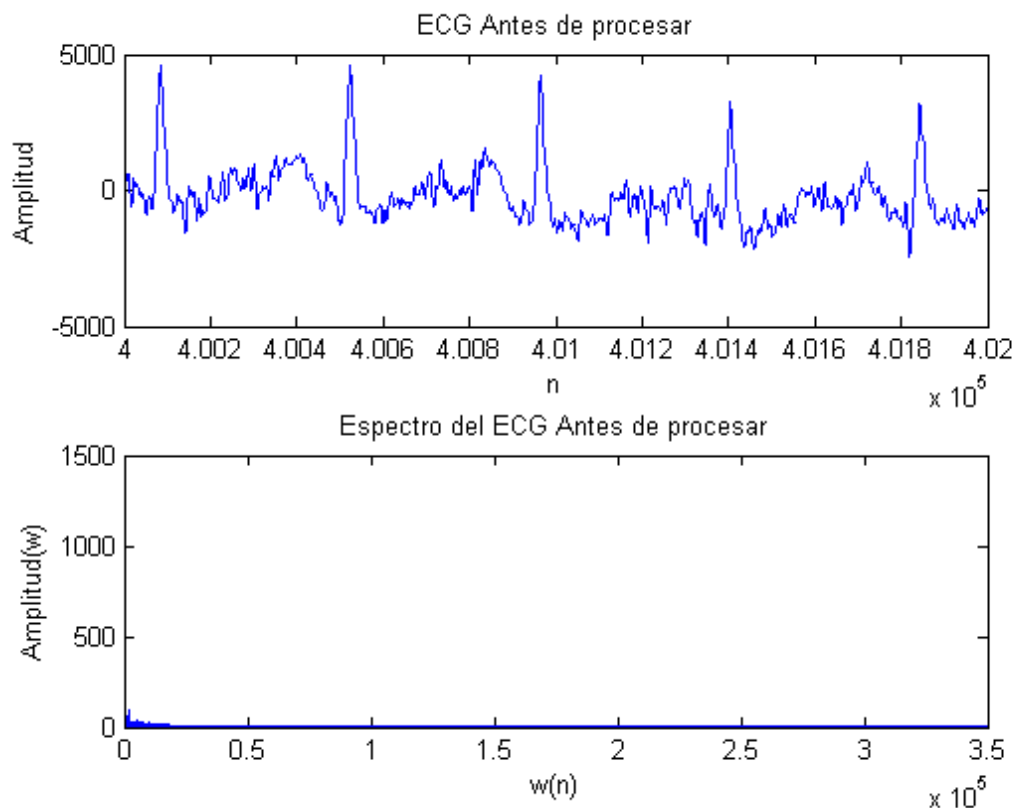


Filtrar una señal de ECG con los filtros nombrados forma parte del procesamiento típico para cualquier análisis automático de la señal electrocardiográfica.

A partir de las señales ECG1.5 provistas por la cátedra se desarrollara un estudio sobre cada señal a partir de los filtros generados obteniendo así diferentes conclusiones para cada una de las mismas.

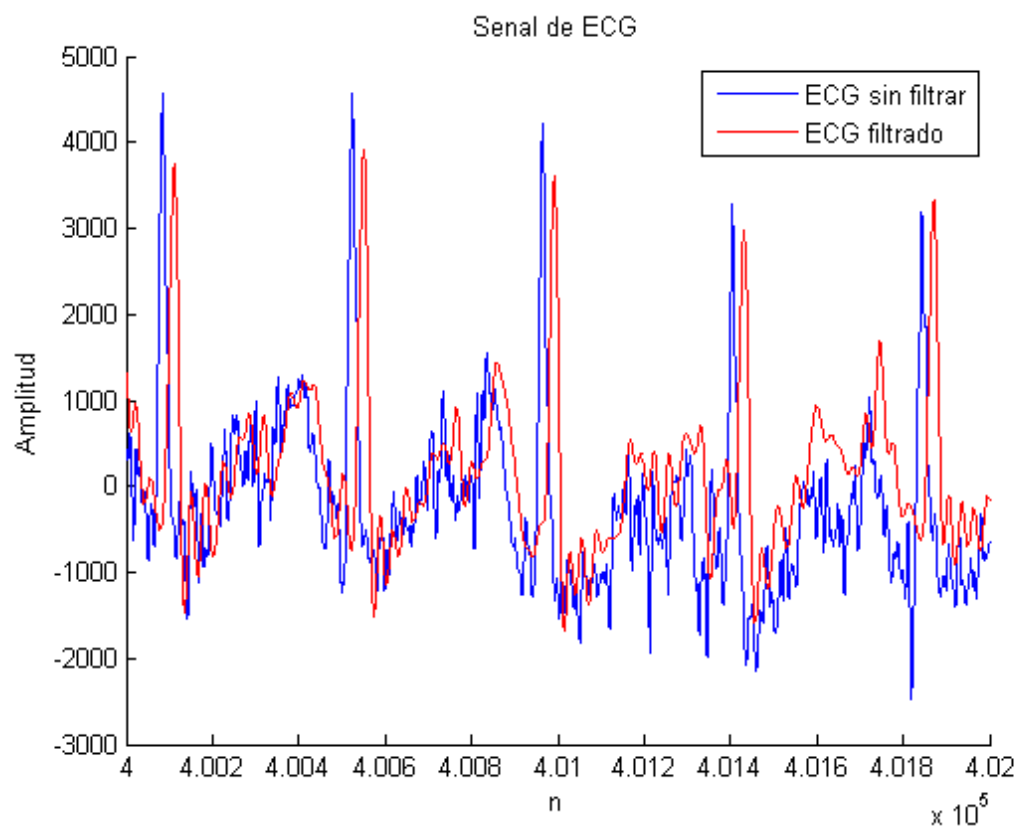
### Estudio para la señal: ECG1

A continuación se expone la señal de ECG con la que se trabajara en este primer estudio. Es de importancia destacar que se realizó un zoom sobre dicha señal de forma de poder visualizar de forma correcta la forma de señal y que este zoom se aplica a todas las señales que se encuentran a continuación. En el caso que sea necesario observar dicha señal completa se podrá observar en la carpeta “señales” adjunta al informe presentado.



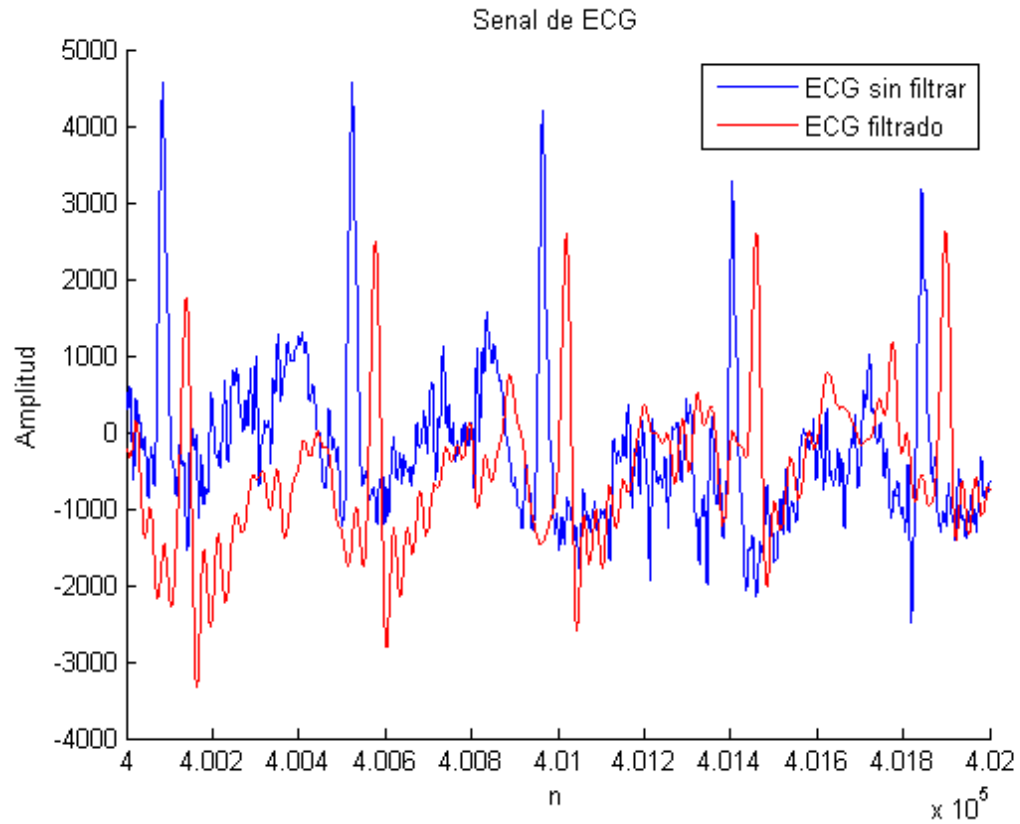
A continuación se muestran los resultados al aplicar los filtros generados a la señal de ECG1 expuesta anteriormente obteniendo los siguientes resultados:

Primer filtrado: Filtro Butterworth



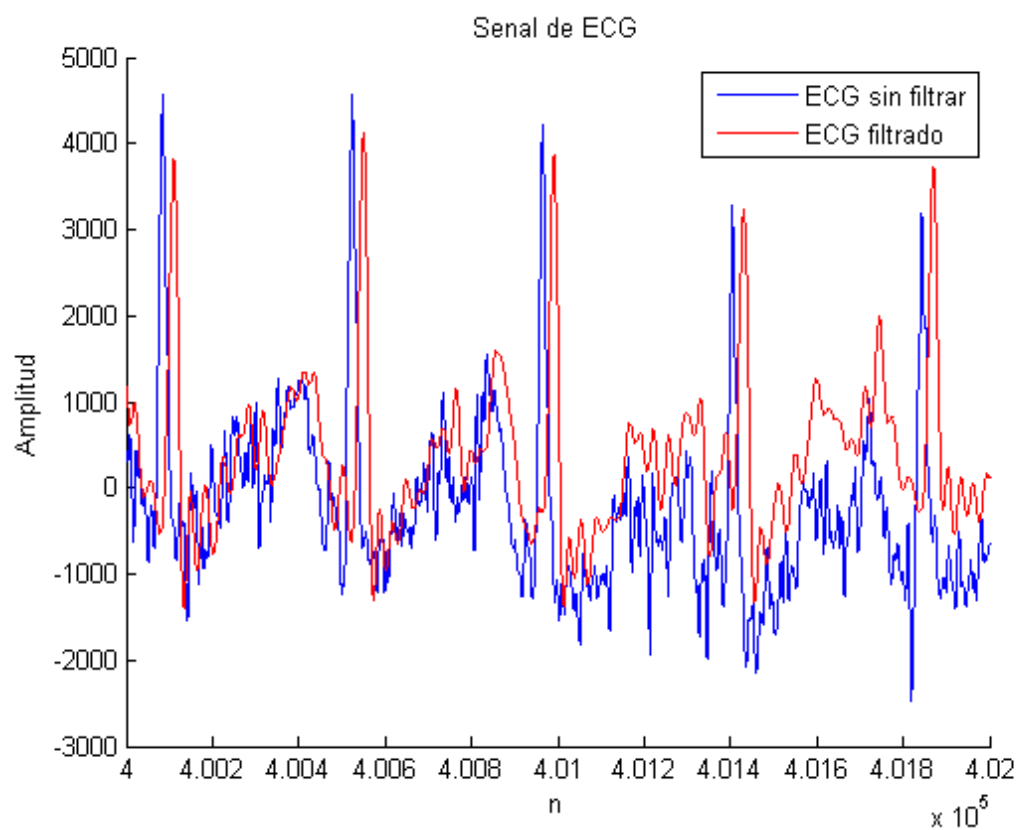
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 1



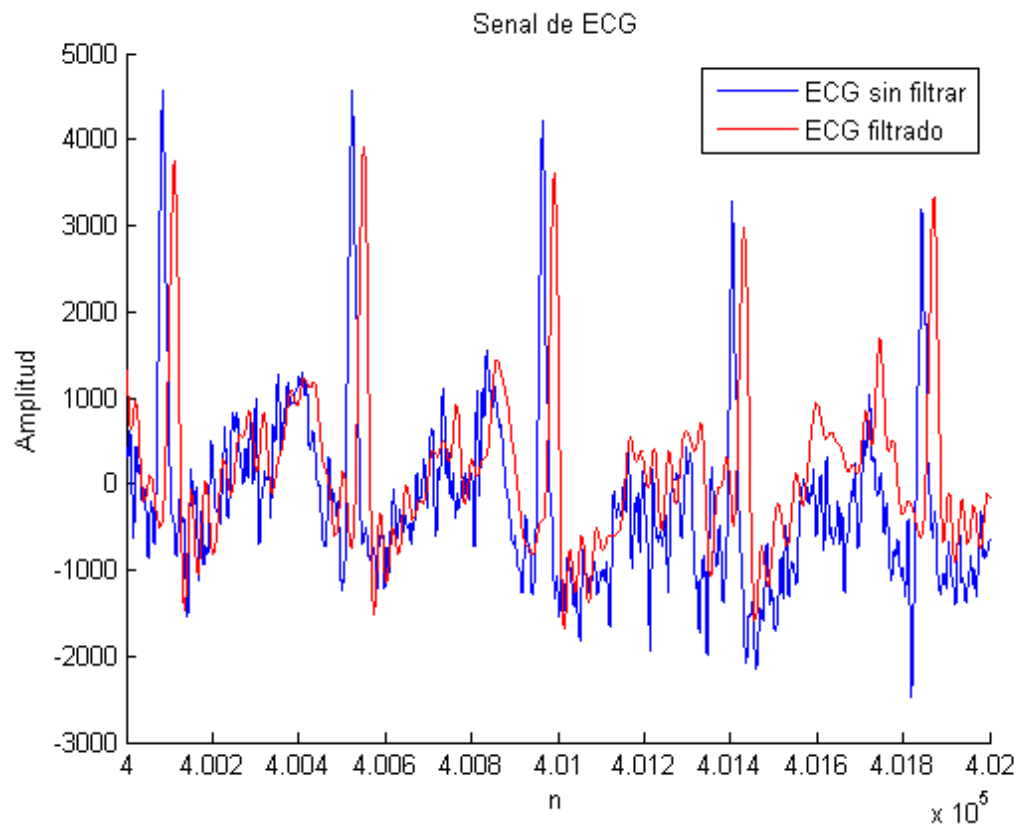
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Tercer filtrado: Filtro Chebyshev 2



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

#### Cuarto filtrado: Filtro Elíptico



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

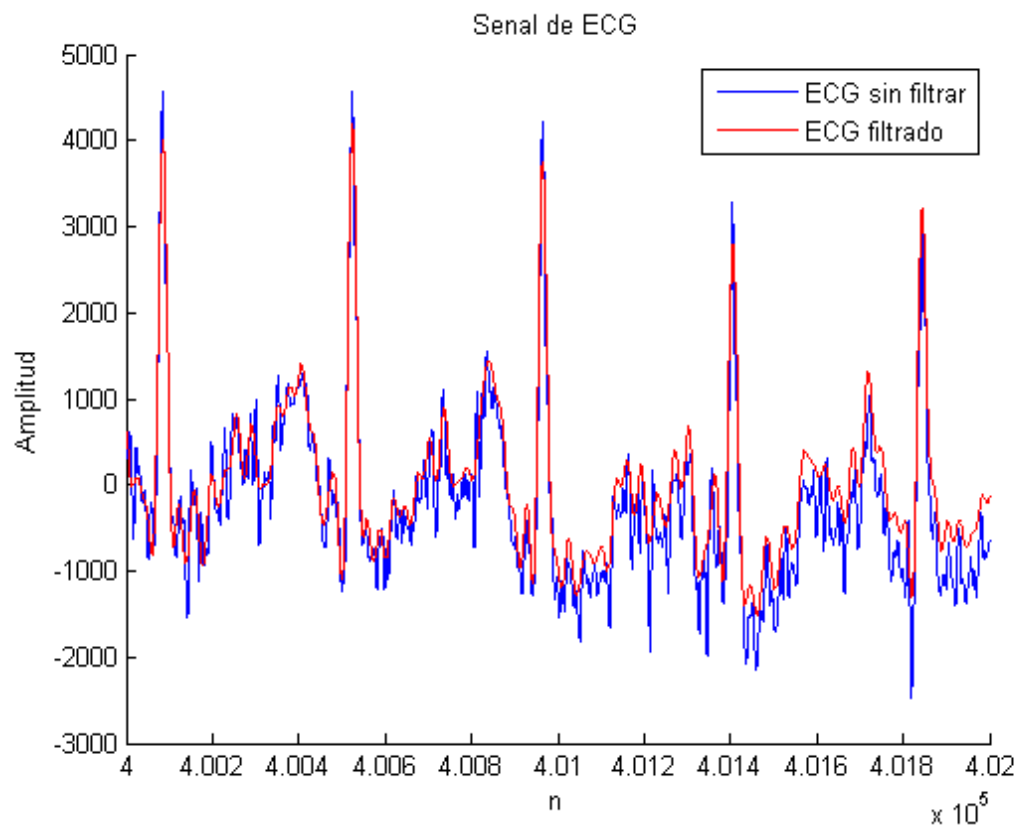


Como se puede ver, de los experimentos realizados, al filtrar la señal de entrada el filtro provoca una distorsión sobre la misma generando un desfase en la señal de salida respecto de la entrada. En ocasiones este desfase puede ser omitido. Debido a que en este caso se trabaja con una señal biológica proveniente de una persona esta distorsión no es permitida.

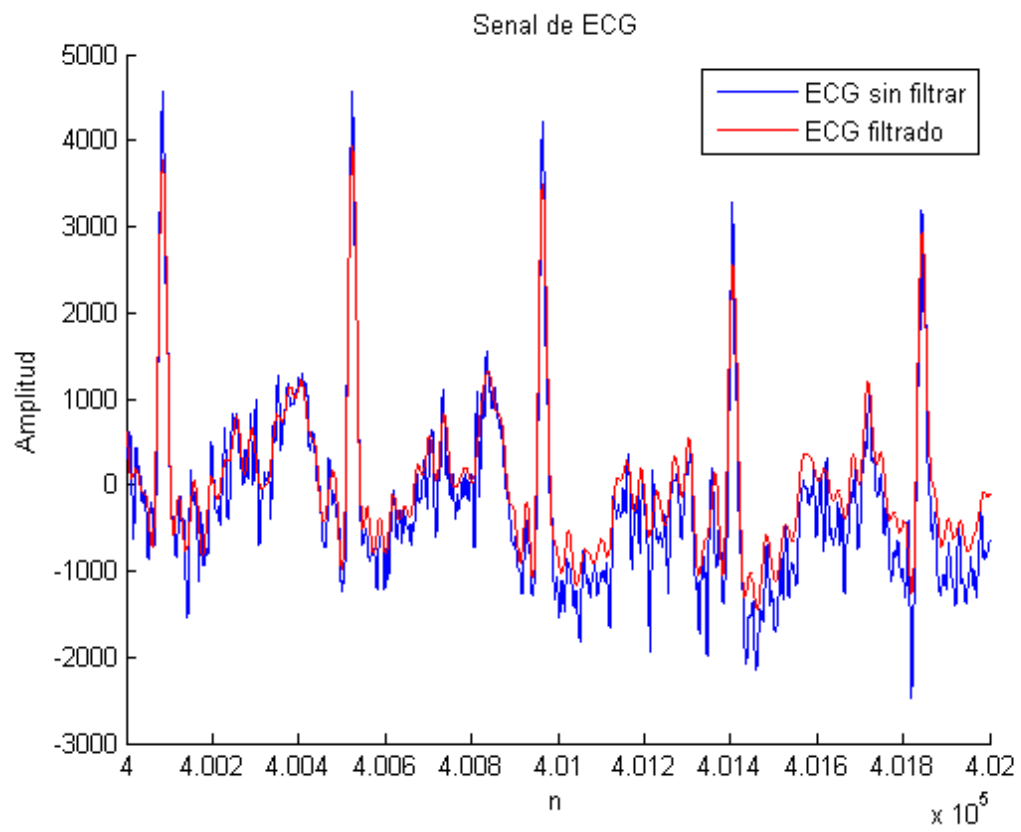
Los filtros IIR no contienen fase lineal, la fase del mismo varía con la frecuencia, motivo por el cual a diferencia de cuando se realizaba el filtrado FIR donde la fase si era lineal y la corrección de fase se realizaba a partir del retardo de grupo, en este tipo de filtros la forma de solucionar este desfase es a través de un doble filtrado. Se realiza un filtrado y luego a través de la inversión de la señal (utilizando la función *fliplr* propuesta por Matlab) se vuelve a realizar un nuevo filtrado. De esta forma se logra anular el desfase producido por el primer filtro aplicado a la señal.

A continuación se aplica la técnica antes mencionada y se exponen los resultados obtenidos.

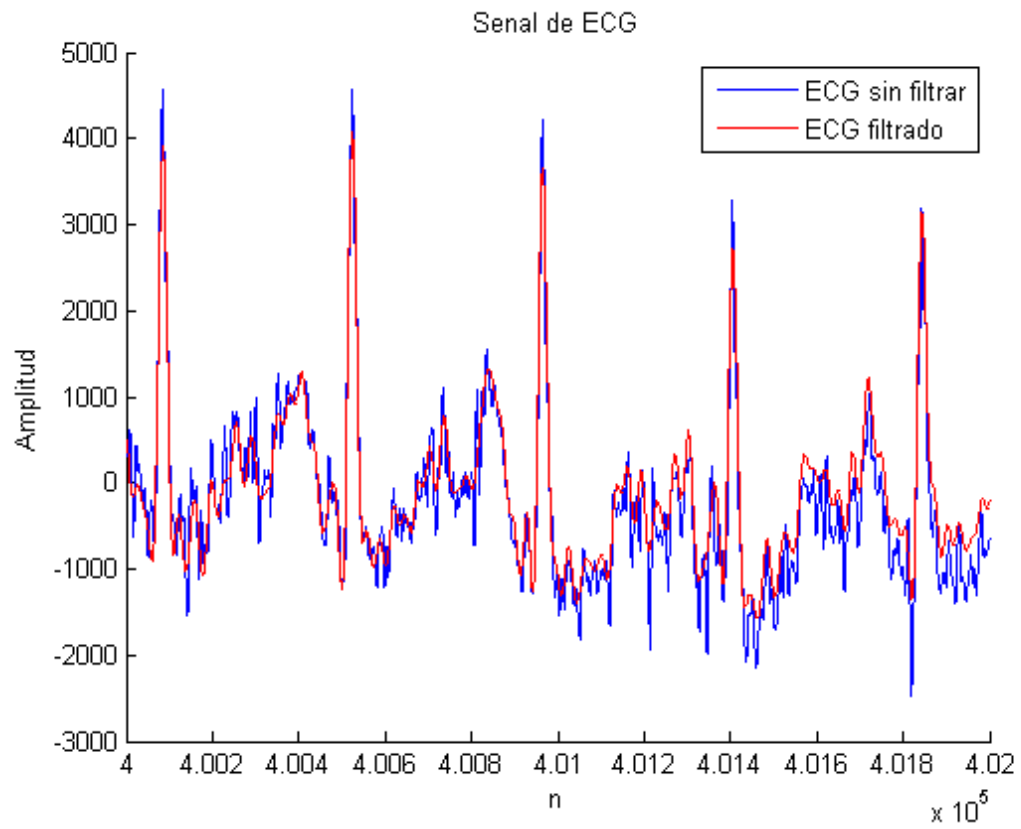
Primer filtrado: Filtro Butterworth (doble filtrado)



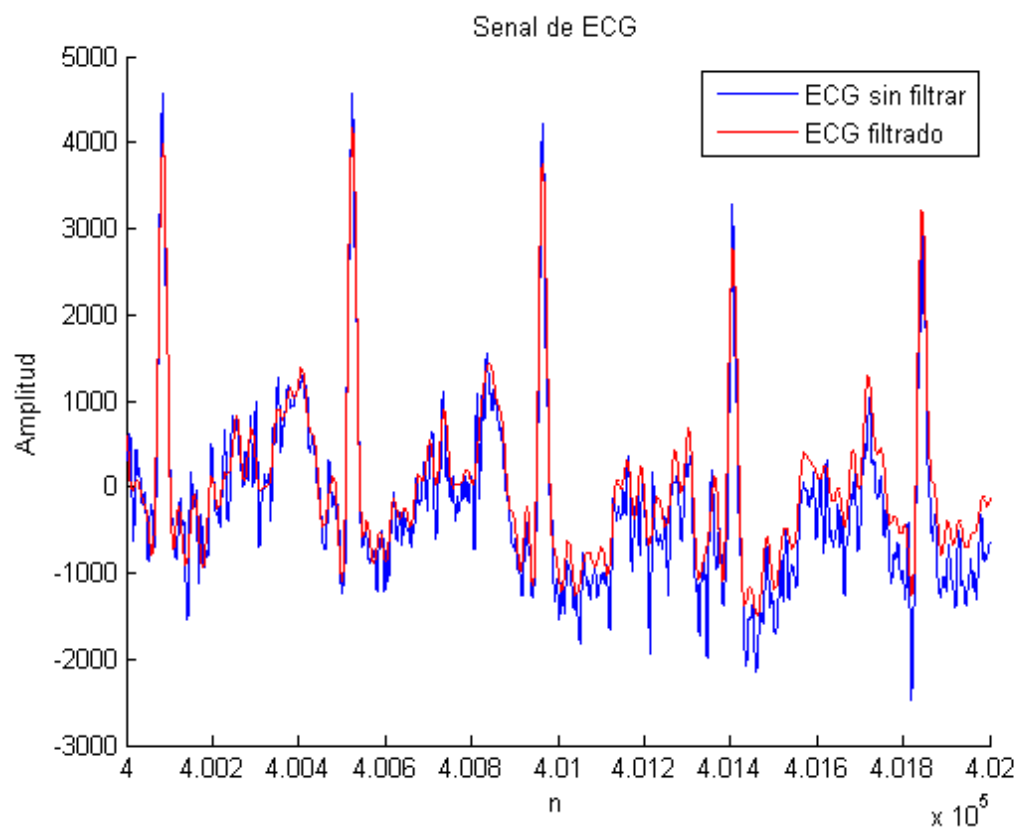
Primer filtrado: Filtro Chebyshev 1 (doble filtrado)



Primer filtrado: Filtro Chebyshev 2 (doble filtrado)



Primer filtrado: Filtro Elíptico (doble filtrado)



Se puede observar en los gráficos expuestos que en los cuatro filtrados realizados sobre la señal de ECG la señal a la salida del sistema se obtuvo en fase con la señal de entrada al filtro motivo por el cual se puede concluir en que el método mencionado y luego utilizado fue efectivo. Con este método entonces, junto a la técnica vista para los filtros FIR en el trabajo práctico 4, se completa entonces dos formas efectivas de corregir la fase cuando se utiliza filtrado digital.

La segunda parte del estudio consiste en verificar si la forma de procesamiento realizada sobre la señal de ECG afecta la morfología de las ondas de manera esperada. Para poder realizar el estudio nos basamos en el cálculo de la energía total y la obtenida luego de filtrar la señal de ECG. Este estudio fue realizado con un filtro Butterworth aunque las conclusiones obtenidas responden a los cuatro filtrados correspondientes al procesamiento típico tanto con, como sin, corrección de fase.

```
EnergiaTotal = sum(abs(fft(ECG1)));
EnergiaConProc = sum(abs(fft(ECG1_filter)));
EnergiaPerdida = EnergiaTotal - EnergiaConProc;
EnergiaPerdida = 100 * (EnergiaPerdida/EnergiaTotal);
```

A continuación se exponen las energías obtenidas, es importante tener en cuenta que en este caso se utilizó la señal completa ECG1 brindada por la cátedra.

Energía total del ECG	1.8796e+11 J
Energía total del ECG luego de procesar	1.2440e+11 J
Energía perdida al procesar la señal	6.3565e+10 J
Porcentaje de energía perdido	% 33.8180

De los valores obtenidos se puede ver que luego de filtrar la señal se obtuvo una pérdida de energía de aproximadamente el %33 de la energía total de la señal de ECG. Una forma de mitigar este problema se expone a continuación.

Podemos observar una pérdida considerable de energía debido a que el filtro pasabanda generado elimina una banda de frecuencias que contiene información de la señal de ECG. Una forma de mitigar este problema es a partir de la modificación de las frecuencias de corte y paso del filtro.

Se generó un nuevo filtro pasabanda cuyas modificaciones respecto al primer filtro generado se encuentran en la Fpass2 y en la Fstop2.

```
Fstop1 = 0.001;      % First Stopband Frequency
Fpass1 = 0.01;       % First Passband Frequency
Fpass2 = 100;        % Second Passband Frequency
Fstop2 = 120;        % Second Stopband Frequency
Astop1 = 60;         % First Stopband Attenuation (dB)
Apass  = 1;          % Passband Ripple (dB)
Astop2 = 80;         % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Porcentaje de energía perdido	% 6.8590
-------------------------------	----------

Con este filtro podemos observar que la energía perdida luego del filtrado es de aproximadamente %7 respecto de la energía de la señal de entrada al filtro.

Como tercera parte del estudio sobre la señal de ECG1 se modificaron las frecuencias de corte superior e inferior hasta comenzar a afectar la morfología de las ondas PQRST.

```
% Señal de ECG general.
```

```
Fstop1 = 1;           % First Stopband Frequency
Fpass1 = 1.5;         % First Passband Frequency
Fpass2 = 40;          % Second Passband Frequency
Fstop2 = 45;          % Second Stopband Frequency
Astop1 = 60;          % First Stopband Attenuation (dB)
Apass  = 1;           % Passband Ripple (dB)
Astop2 = 80;          % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

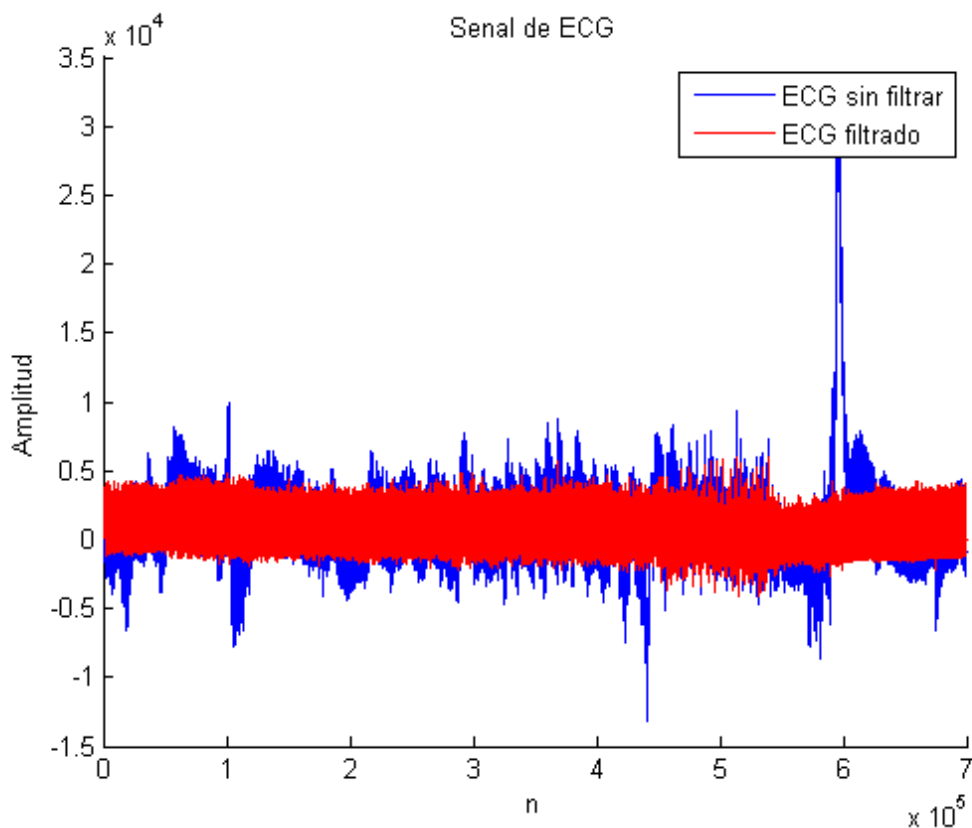
% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Después de realizar un proceso de iteración intercambiando las frecuencias indicadas se pudo estimar que con las frecuencias expuestas a continuación la señal comienza a cambiar su morfología.

Fstop1	Fpass1	Fpass2	Fstop2
1 Hz	1.5 Hz	40 Hz	45 Hz



Señal obtenida a la salida del filtro:



Con lo cual se puede concluir en que para la visualización de la señal de ECG se requiere de un ancho de banda de 1.5Hz a 40Hz para que la misma pueda visualizarse de forma correcta. Es importante destacar que en este rango de frecuencias no se encuentra toda la información de la señal de ECG motivo por el cual el realizar un estudio de patologías en dicho rango de frecuencias resultaría erróneo.

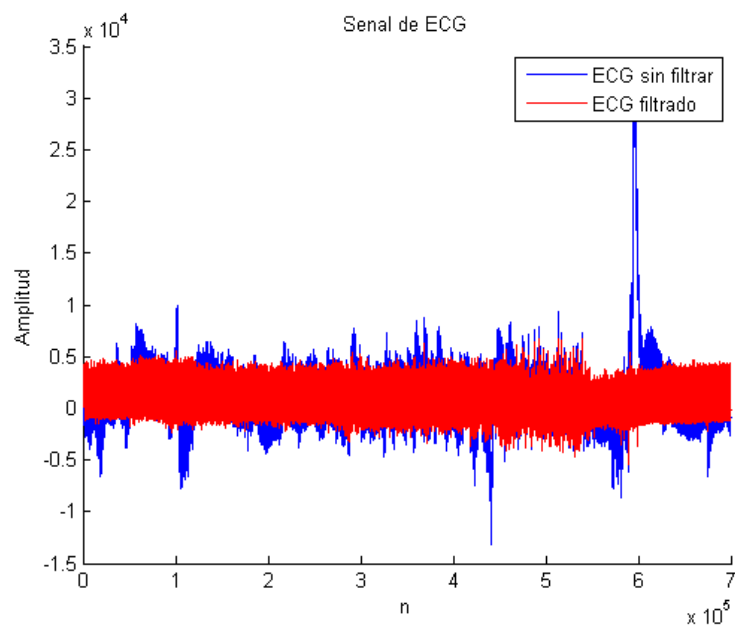
En esta cuarta etapa del estudio se busca filtrar los movimientos bruscos de línea. Nuevamente “a prueba y error” se fueron variando las frecuencias de corte del filtro pasabanda hasta que la señal de línea de base se anuló casi por completo.

```
% Línea de base.

Fstop1 = 1;           % First Stopband Frequency
Fpass1 = 1.1;         % First Passband Frequency
Fpass2 = 100;         % Second Passband Frequency
Fstop2 = 120;         % Second Stopband Frequency
Astop1 = 60;          % First Stopband Attenuation (dB)
Apass  = 1;           % Passband Ripple (dB)
Astop2 = 80;          % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

A continuación se puede observar los resultados luego de filtrar la señal.



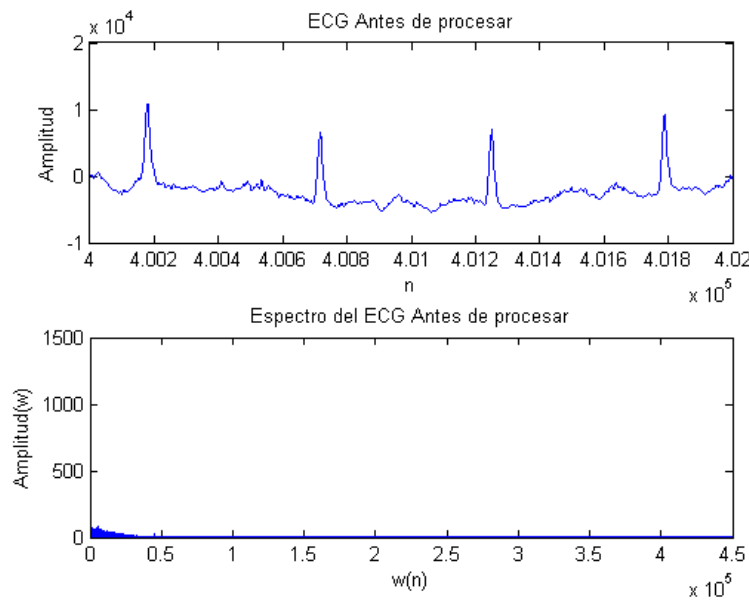
Se puede notar en el gráfico expuesto que la señal de línea de base no desapareció por completo, si bien esta se anuló en gran parte de la señal de ECG en otras partes disminuyó pero no llegó a anularse.

## Estudio para la señal: ECG2

A continuación se realiza un estudio similar al realizado para la señal de ECG1 con la señal ECG2. A diferencia del primer estudio realizado para este solo se genera un filtrado sobre la señal con los cuatro filtros pasabanda desarrollados y luego se buscará disminuir la energía de ECG perdida al realizar el filtrado.

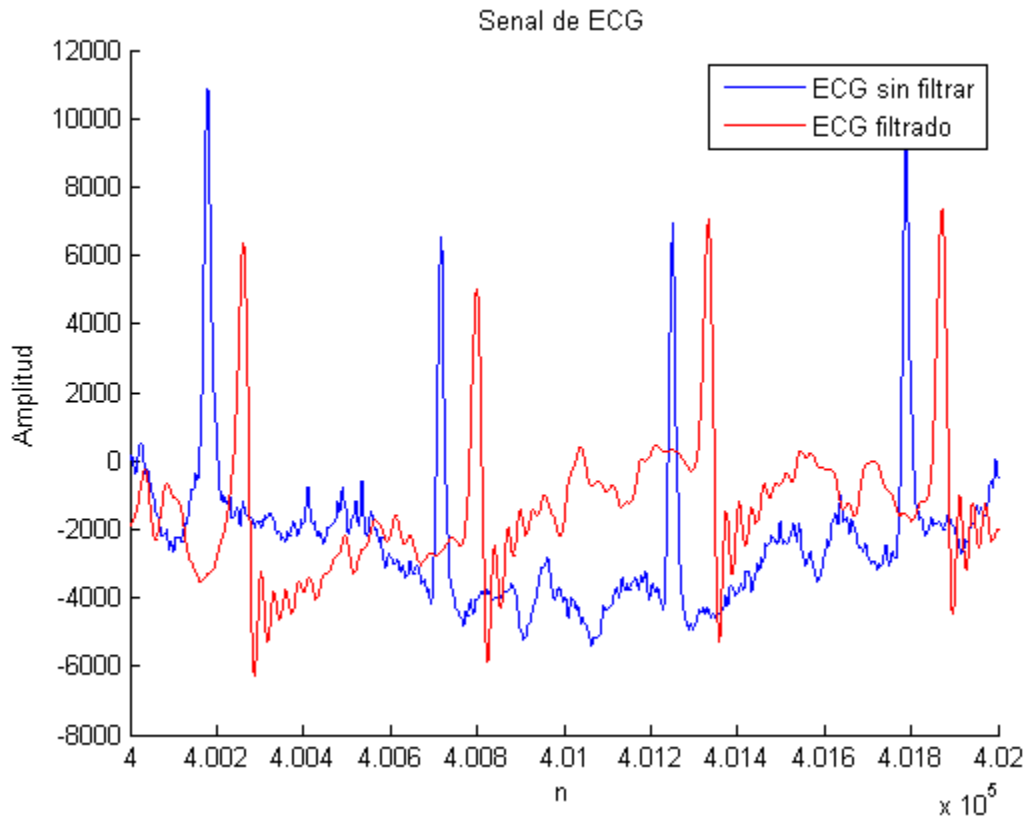
A continuación se expone la señal de ECG con la que se trabajara en este segundo estudio.

```
% Ploteo de señal de ECG - Tiempo
subplot(2,1,1); plot(4e5:4.02e5,ECG1(4e5:4.02e5));
xlabel('n'), ylabel('Amplitud');
title('ECG Antes de procesar');
% Ploteo de señal ECG - Frecuencia
subplot(2,1,2);
ECG1_tf = 2 / length(ECG1) * fft(ECG1);
plot(abs(ECG1_tf(1:length(ECG1_tf)/2)));
xlabel('w(n)'), ylabel('Amplitud(w)');
title('Espectro del ECG Antes de procesar');
```



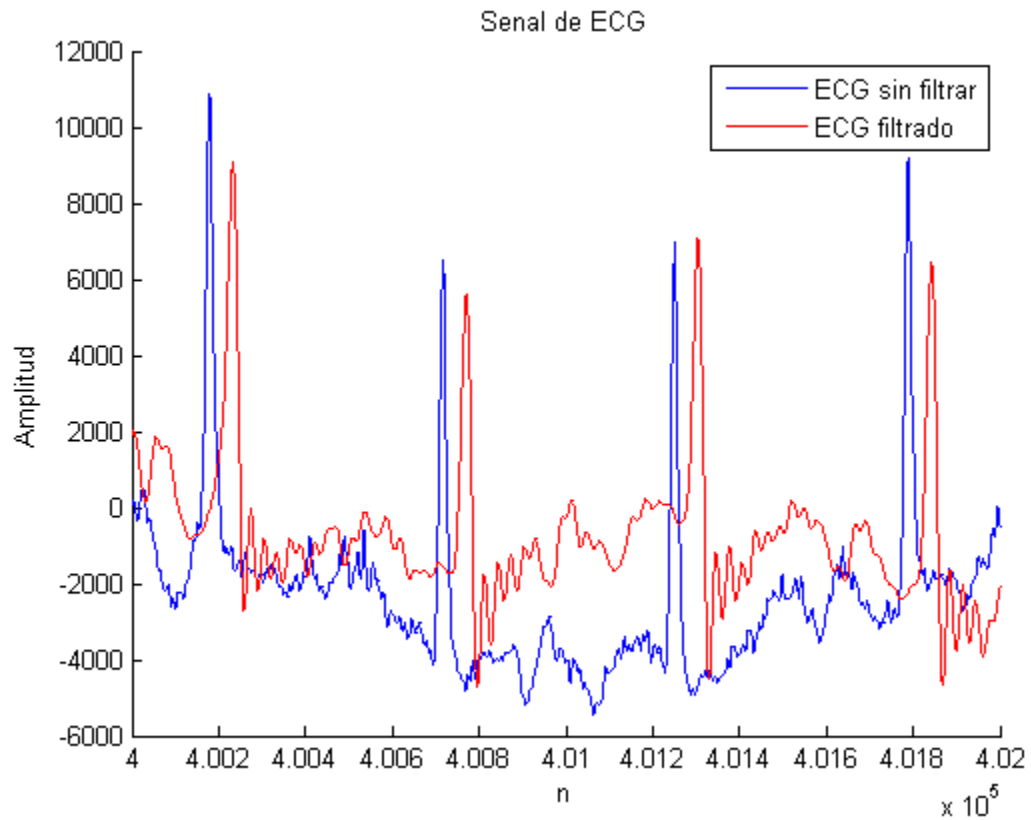
A continuación se muestran los resultados al aplicar los filtros generados a la señal de ECG2 expuesta anteriormente obteniendo los siguientes resultados:

Primer filtrado: Filtro Butterworth



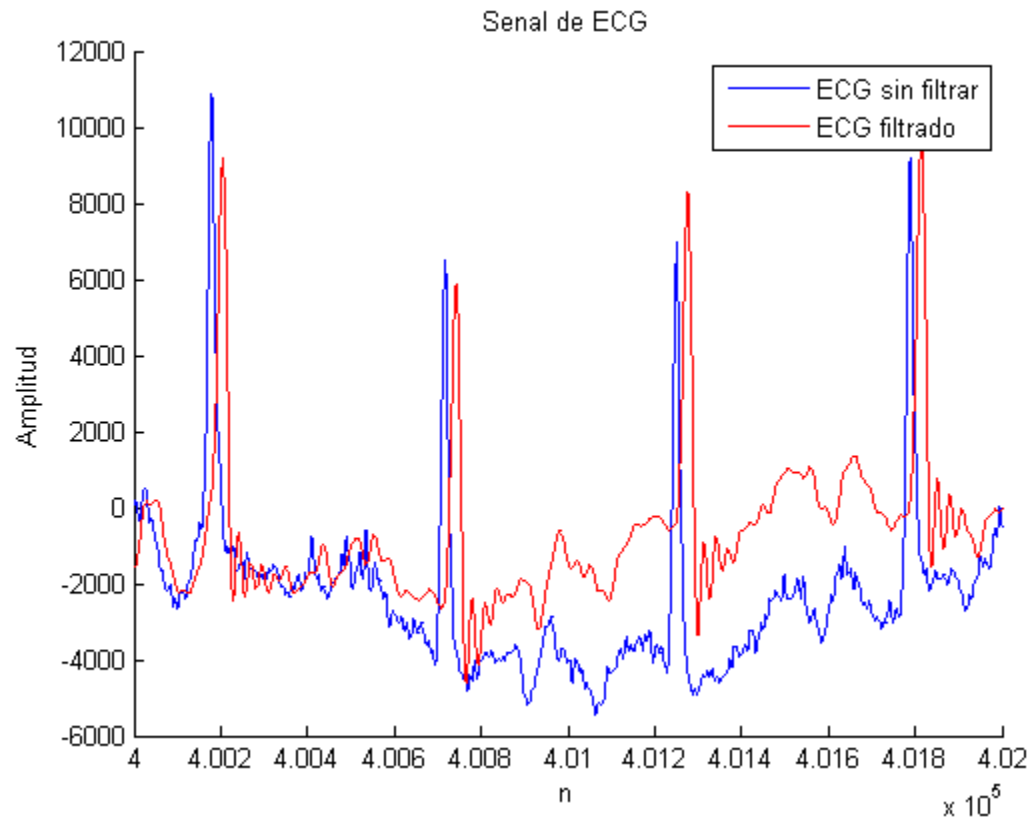
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 1



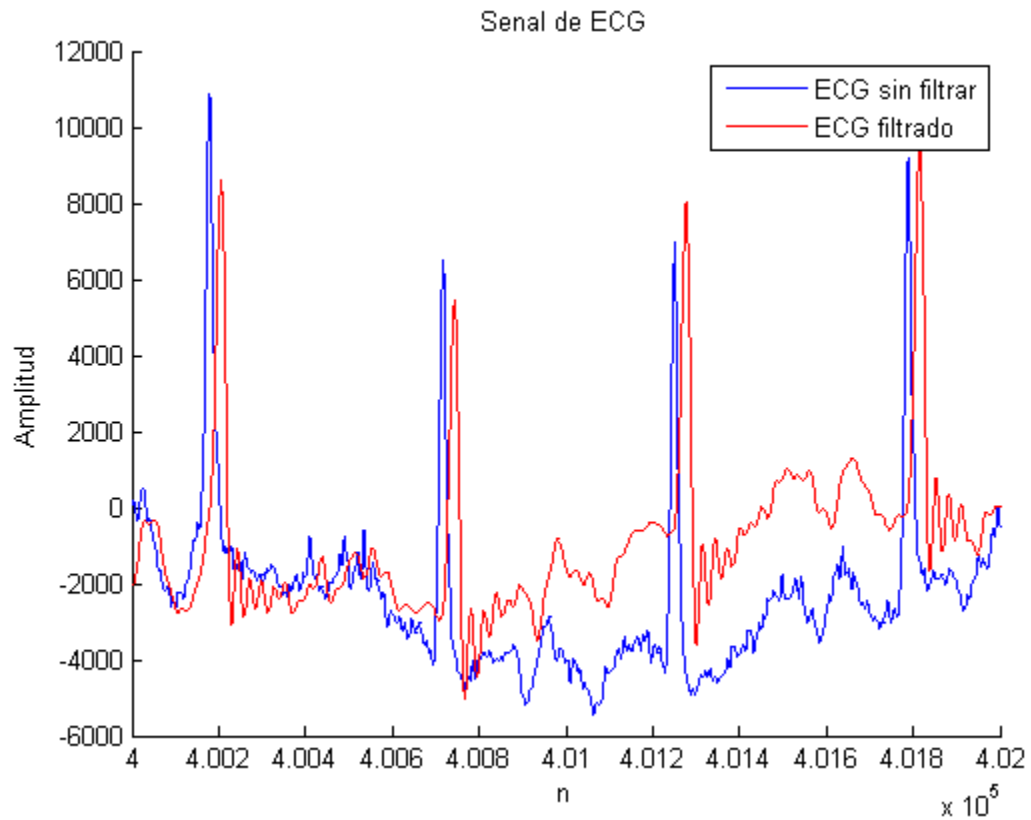
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 3



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

#### Cuarto filtrado: Filtro Elíptico

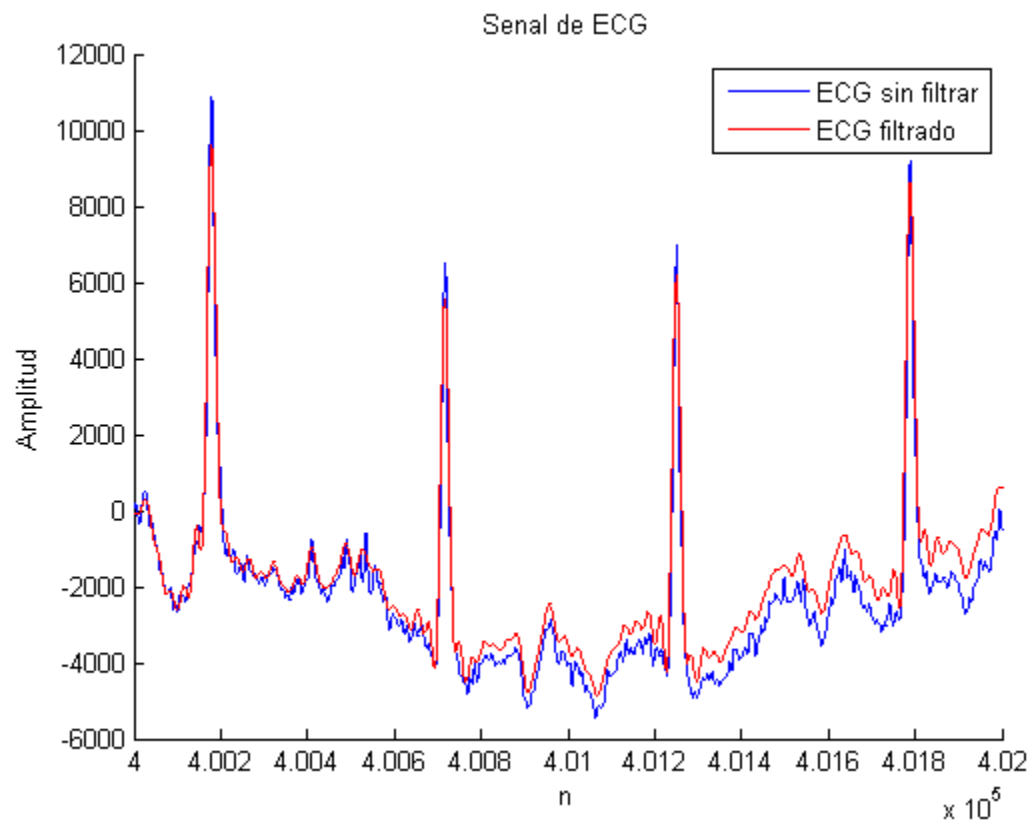


Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

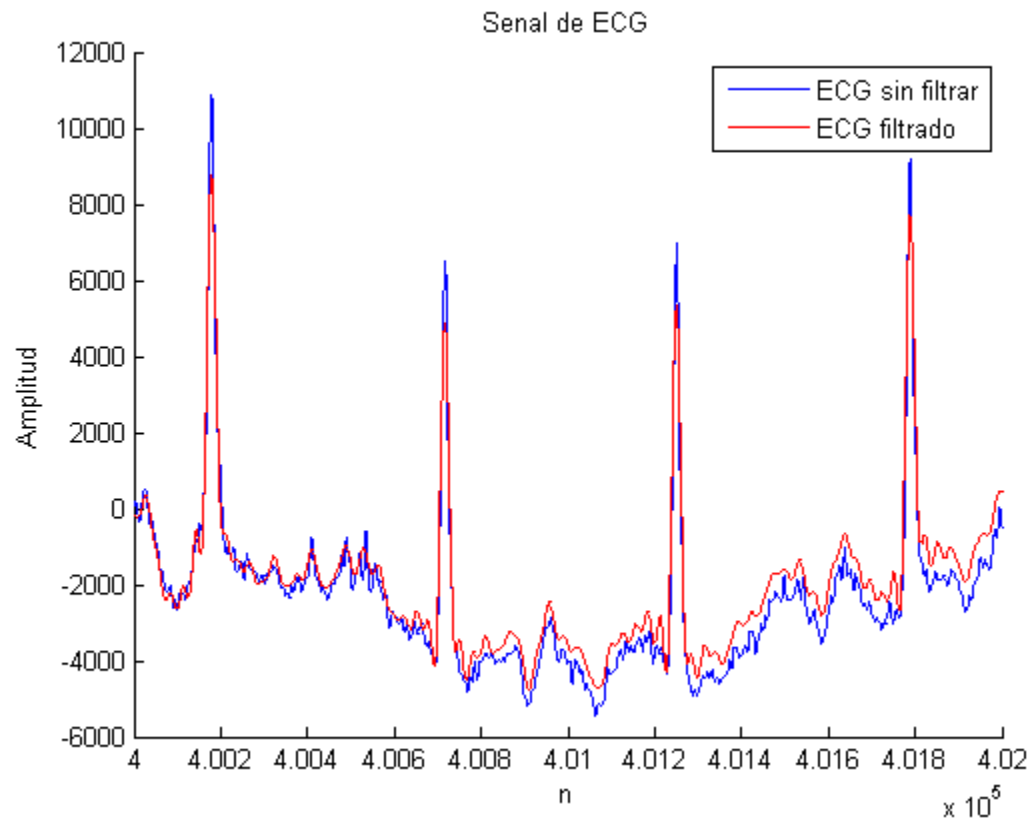


A continuación se aplica la técnica de doble filtrado mencionada cuando se analizó el desfase en la señal ECG1.

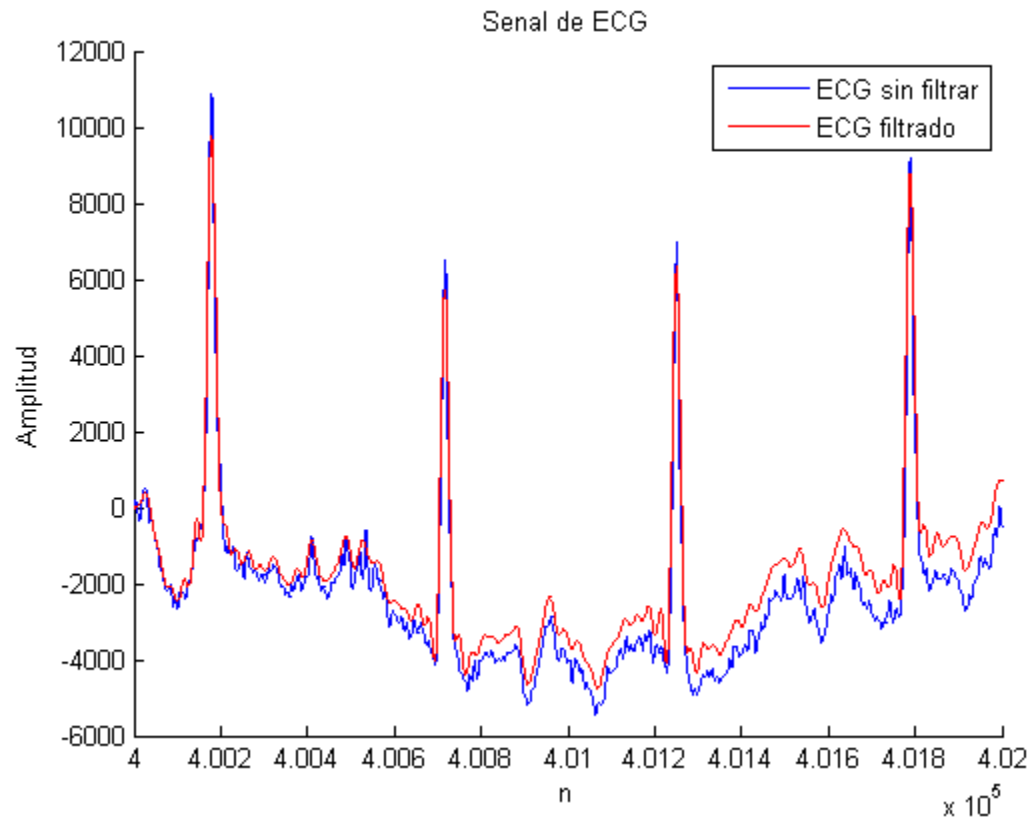
Primer filtrado: Butterworth (doble filtrado)



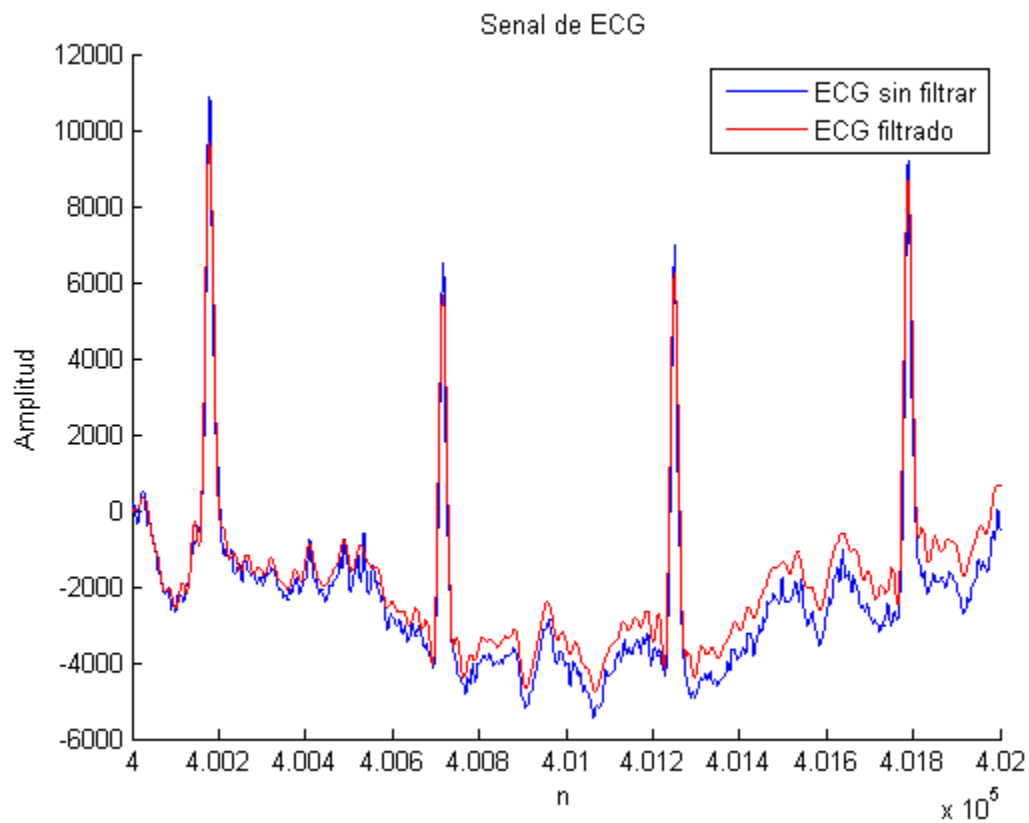
Segundo filtrado: Chebyshev 1 (doble filtrado)



Tercer filtrado: Chebyshev 2 (doble filtrado)



Cuarto filtrado: Eliptico (doble filtrado)



Se puede observar en los gráficos expuestos que en los cuatro filtrados realizados sobre la señal de ECG la señal a la salida del sistema se obtuvo en fase con la señal de entrada al filtro motivo por el cual se puede concluir en que el método mencionado y luego utilizado fue efectivo al igual que cuando se trabajó con la señal ECG1.

A continuación realizaremos un segundo estudio igual al ya realizado para la señal ECG1 utilizando el mismo tipo de filtro sin corrección de fase.

```
EnergiaTotal = sum(abs(fft(ECG1)));  
EnergiaConProc = sum(abs(fft(ECG1_filter)));  
EnergiaPerdida = EnergiaTotal - EnergiaConProc;  
EnergiaPerdida = 100 * (EnergiaPerdida/EnergiaTotal);
```

Energía total del ECG	4.5906e+11 J
Energía total del ECG luego de procesar	3.5817e+11 J
Energía perdida al procesar la señal	1.0089e+11 J
Porcentaje de energía perdido	% 21.9767

De los valores obtenidos se puede observar para la señal ECG2 que luego de filtrar la misma se obtuvo una perdida de energia de aproximadamente el %22 de la energía total de la señal de ECG. A continuación nuevamente se busca minimizar dicha pérdida de energía.

Se generó un filtro pasa banda con las mismas frecuencias de corte utilizadas para el tercer estudio sobre la señal ECG1

```
Fstop1 = 0.001;      % First Stopband Frequency
Fpass1 = 0.01;       % First Passband Frequency
Fpass2 = 100;        % Second Passband Frequency
Fstop2 = 120;        % Second Stopband Frequency
Astop1 = 60;         % First Stopband Attenuation (dB)
Apass  = 1;          % Passband Ripple (dB)
Astop2 = 80;         % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Porcentaje de energía perdido	% 3.7106
-------------------------------	----------

Con este filtro podemos observar que la energía perdida del filtrado es de aproximadamente %7 respecto de la energía de la señal de entrada al filtro.

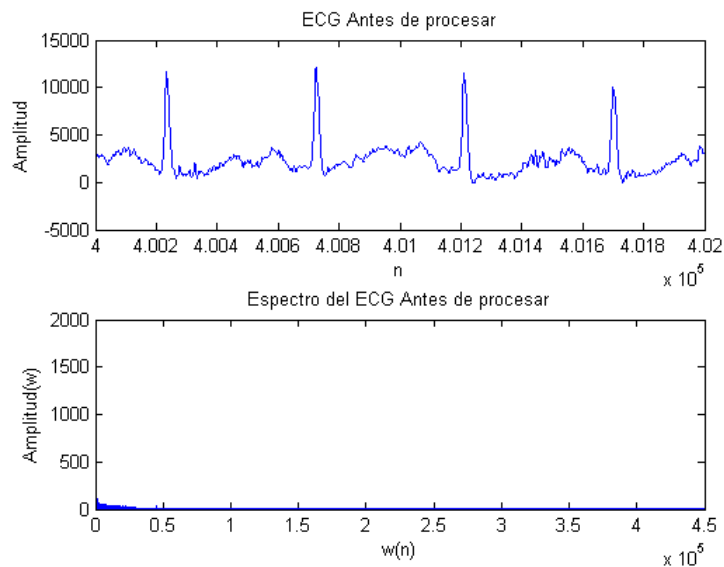
## Estudio para la señal: ECG3

A continuación se realiza el mismo estudio realizado para la señal ECG2 con la señal ECG3.

A continuación se expone la señal de ECG con la que se trabajara en este segundo estudio.

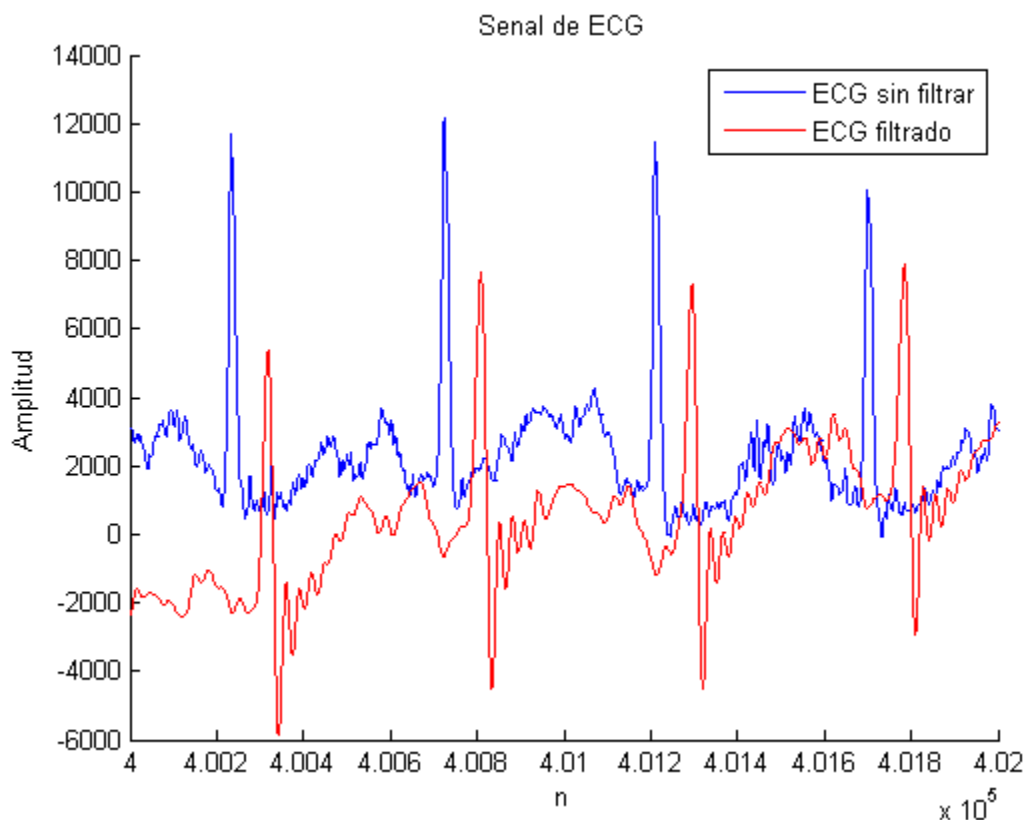
```
% Ploteo de señal de ECG - Tiempo
subplot(2,1,1); plot(4e5:4.02e5,ECG1(4e5:4.02e5));
xlabel('n'), ylabel('Amplitud');
title('ECG Antes de procesar');
% Ploteo de señal ECG - Frecuencia

subplot(2,1,2);
ECG1_tf = 2 / length(ECG1) * fft(ECG1);
plot(abs(ECG1_tf(1:length(ECG1_tf)/2)));
xlabel('w(n)'), ylabel('Amplitud(w)');
title('Espectro del ECG Antes de procesar');
```



A continuación se muestran los resultados al aplicar los filtros generados a la señal de ECG3 expuesta anteriormente obteniendo los siguientes resultados:

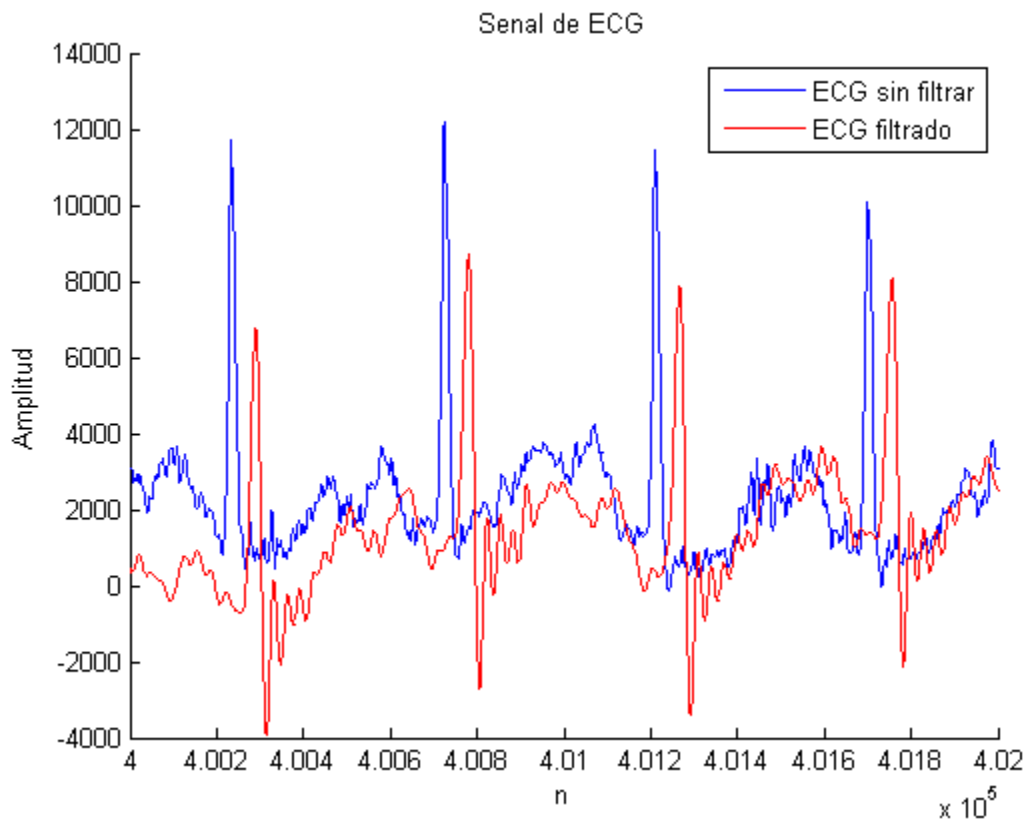
Primer filtrado: Filtro Butterworth



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

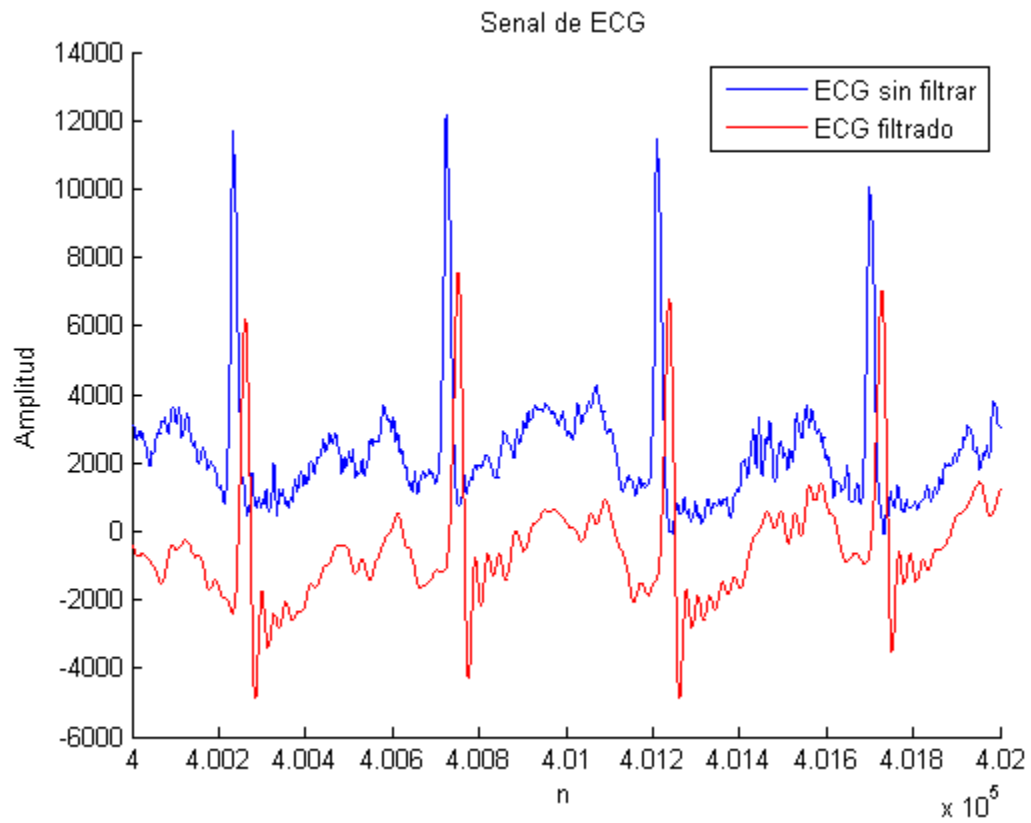


Segundo filtrado: Filtro Chebyshev 1



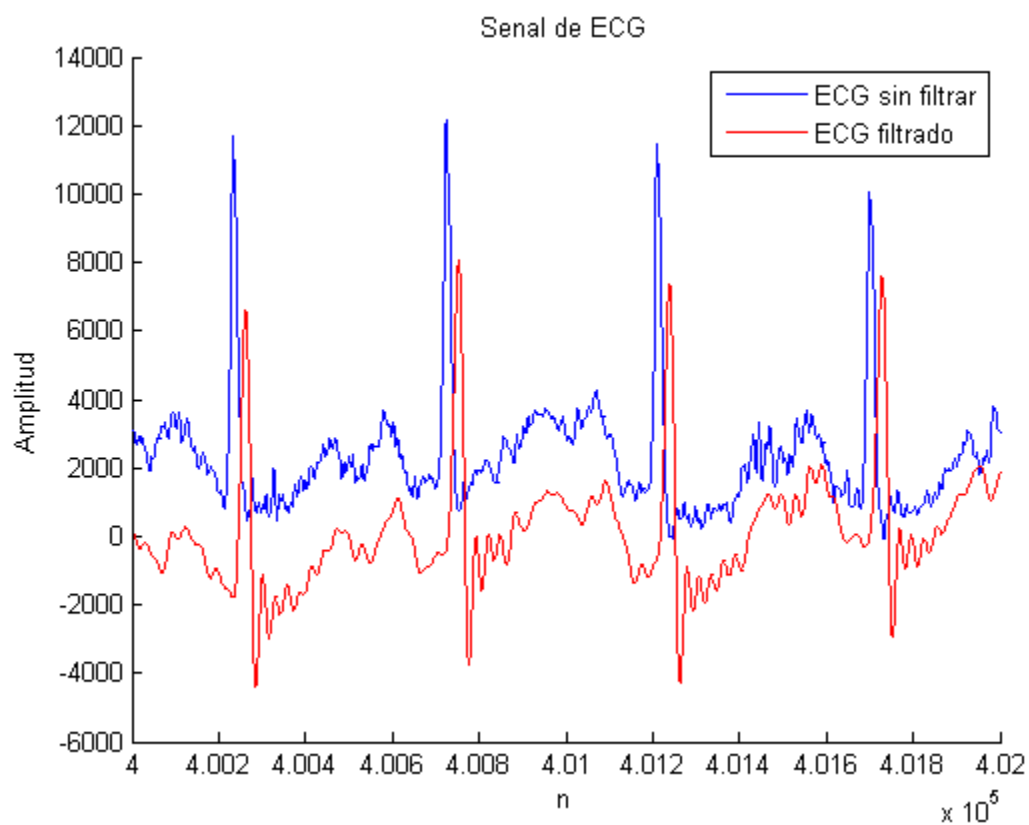
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 2



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

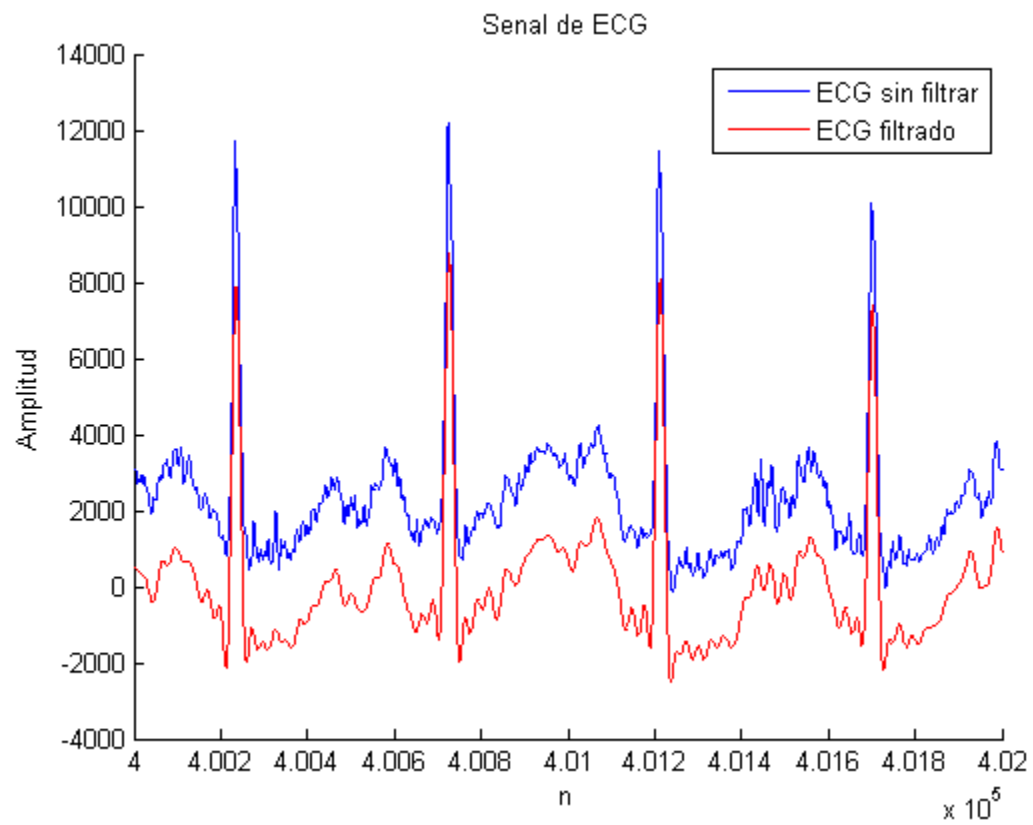
#### Cuarto filtrado: Filtro Elíptico



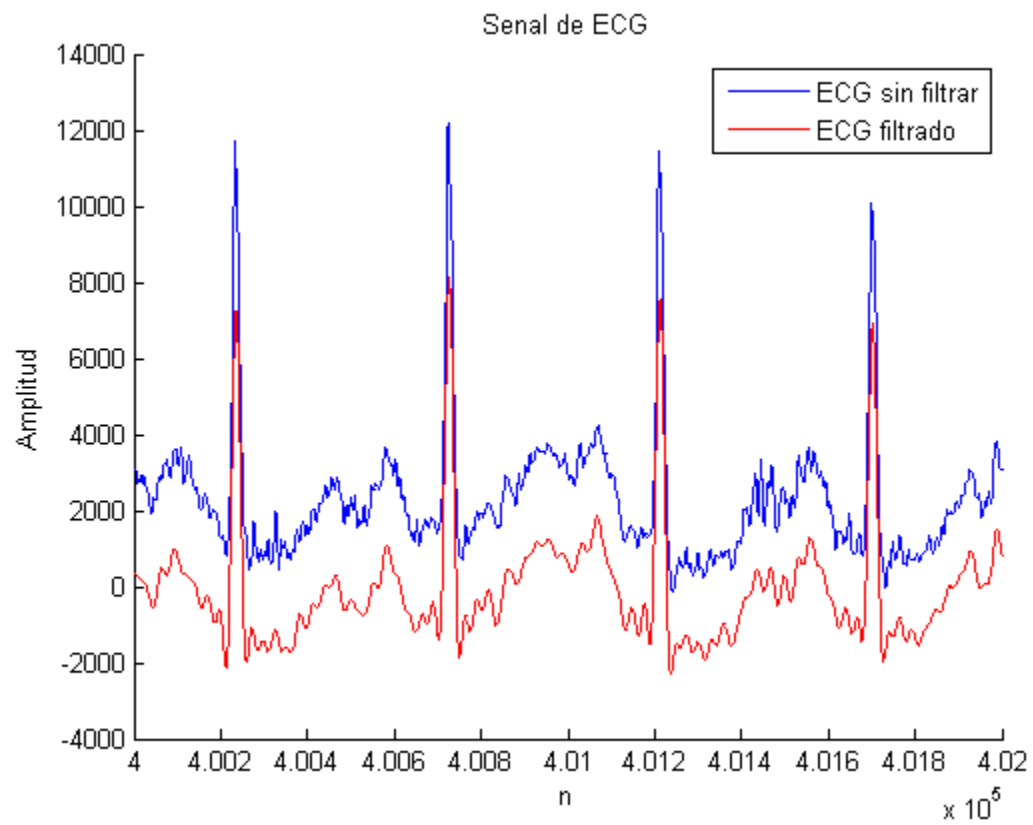
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

A continuación se aplica la técnica de doble filtrado mencionada cuando se analizó el desfase en la señal ECG3.

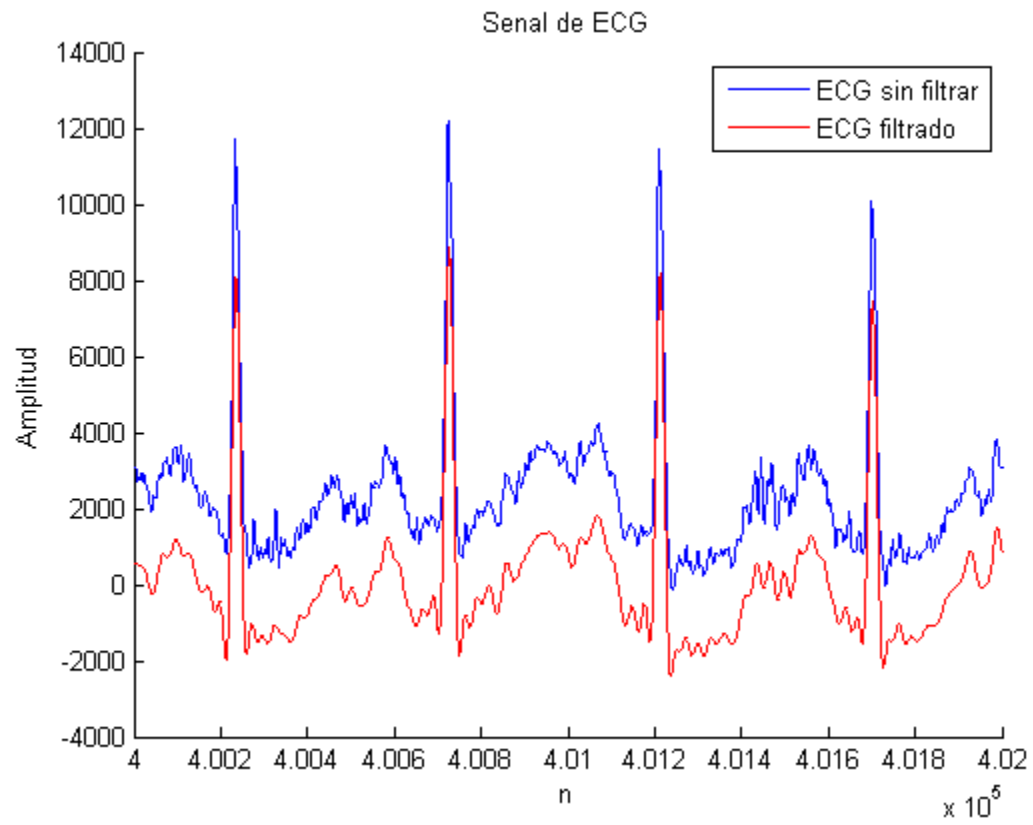
Primer filtrado: Butterworth (doble filtrado)



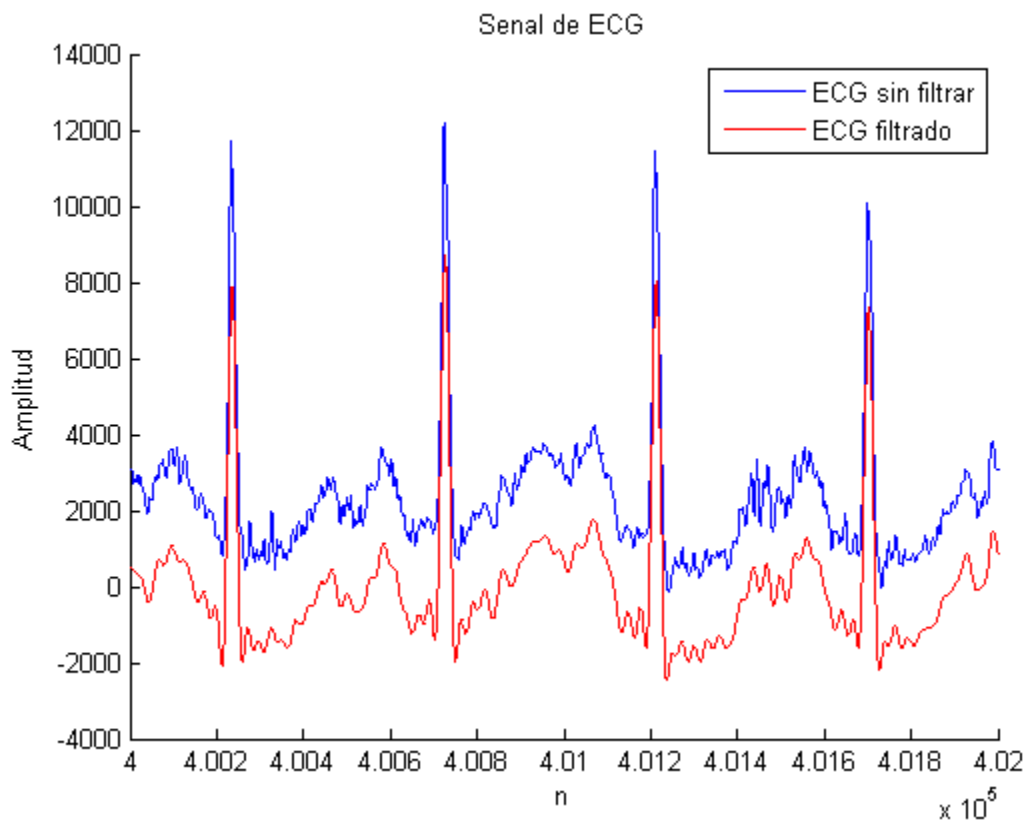
Segundo filtrado: Chebyshev 1 (doble filtrado)



Tercer filtrado: Chebyshev 2 (doble filtrado)



Cuarto filtrado: Eliptico (doble filtrado)



Se puede observar en los gráficos expuestos que en los cuatro filtrados realizados sobre la señal de ECG la señal a la salida del sistema se obtuvo en fase con la señal de entrada al filtro motivo por el cual se puede concluir en que el método mencionado y luego utilizado fue efectivo al igual que cuando se trabajó con la señal ECG1.

A continuación realizaremos un segundo estudio igual al ya realizado para la señal ECG1 utilizando el mismo tipo de filtro sin corrección de fase.

```
EnergiaTotal = sum(abs(fft(ECG1)));
EnergiaConProc = sum(abs(fft(ECG1_filter)));
EnergiaPerdida = EnergiaTotal - EnergiaConProc;
EnergiaPerdida = 100 * (EnergiaPerdida/EnergiaTotal);
```

Energía total del ECG	5.0543e+11 J
Energía total del ECG luego de procesar	3.6747e+11 J
Energía perdida al procesar la señal	1.3797e+11 J
Porcentaje de energía perdido	% 27.2969

De los valores obtenidos se puede observar para la señal ECG3 que luego de filtrar la misma se obtuvo una perdida de energia de aproximadamente el %27 de la energía total de la señal de ECG. A continuación nuevamente se busca minimizar dicha pérdida de energía.



Se generó un filtro pasa banda con las mismas frecuencias de corte utilizadas para el tercer estudio sobre la señal ECG1

```
Fstop1 = 0.001;      % First Stopband Frequency
Fpass1 = 0.01;       % First Passband Frequency
Fpass2 = 100;        % Second Passband Frequency
Fstop2 = 120;        % Second Stopband Frequency
Astop1 = 60;         % First Stopband Attenuation (dB)
Apass  = 1;          % Passband Ripple (dB)
Astop2 = 80;         % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Porcentaje de energía perdido	% 6.4271
-------------------------------	----------

Con este filtro podemos observar que la energía perdida del filtrado es de aproximadamente %6.5 respecto de la energía de la señal de entrada al filtro.

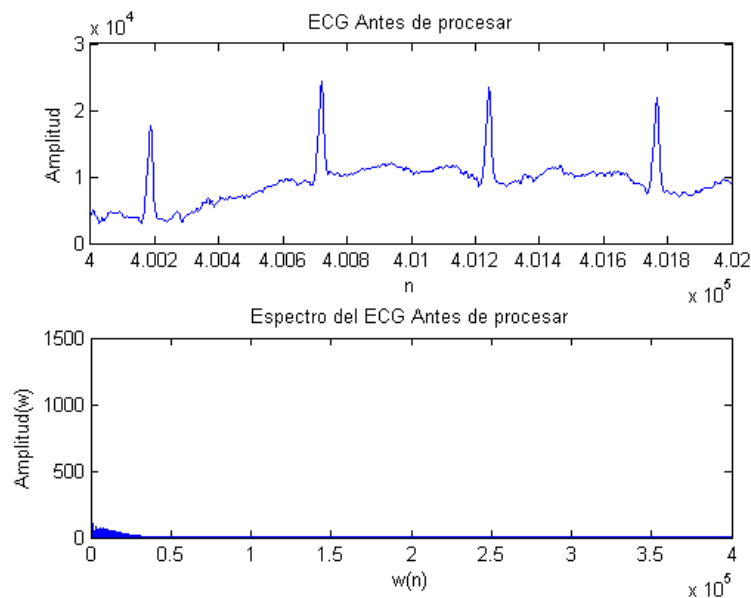
## Estudio para la señal: ECG4

A continuación se realiza el mismo estudio realizado para la señal ECG2 con la señal ECG4.

A continuación se expone la señal de ECG con la que se trabajara en este segundo estudio.

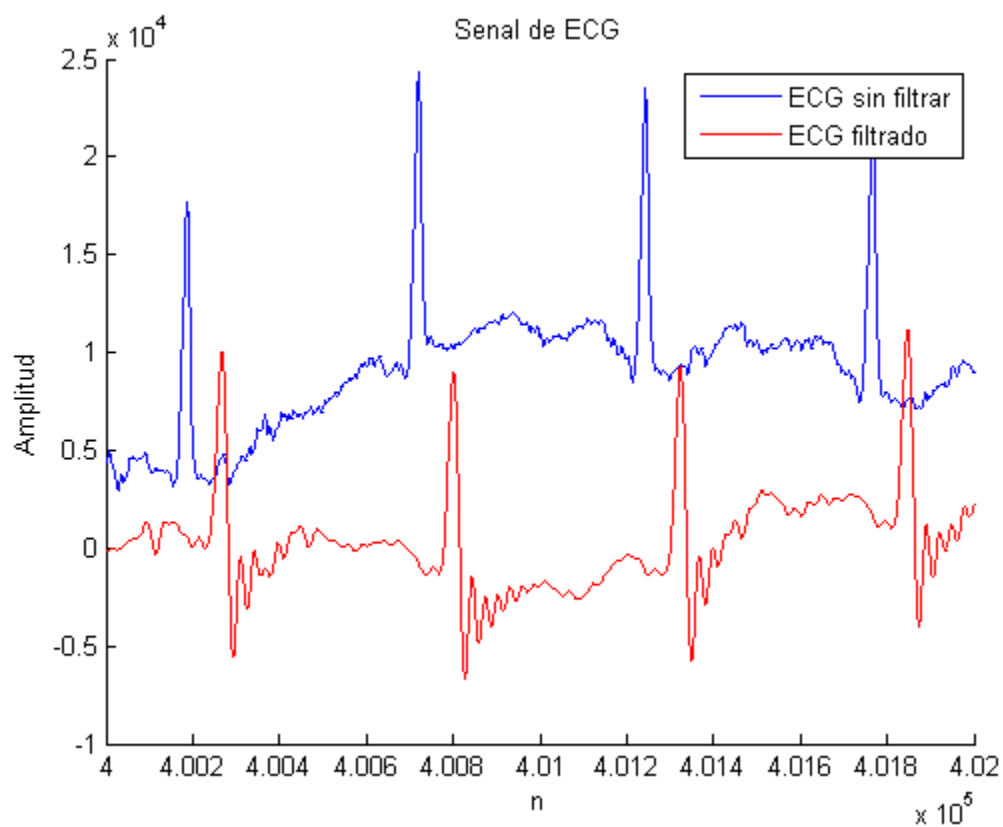
```
% Ploteo de señal de ECG - Tiempo
subplot(2,1,1); plot(4e5:4.02e5,ECG1(4e5:4.02e5));
xlabel('n'), ylabel('Amplitud');
title('ECG Antes de procesar');
% Ploteo de señal ECG - Frecuencia

subplot(2,1,2);
ECG1_tf = 2 / length(ECG1) * fft(ECG1);
plot(abs(ECG1_tf(1:length(ECG1_tf)/2)));
xlabel('w(n)'), ylabel('Amplitud(w)');
title('Espectro del ECG Antes de procesar');
```



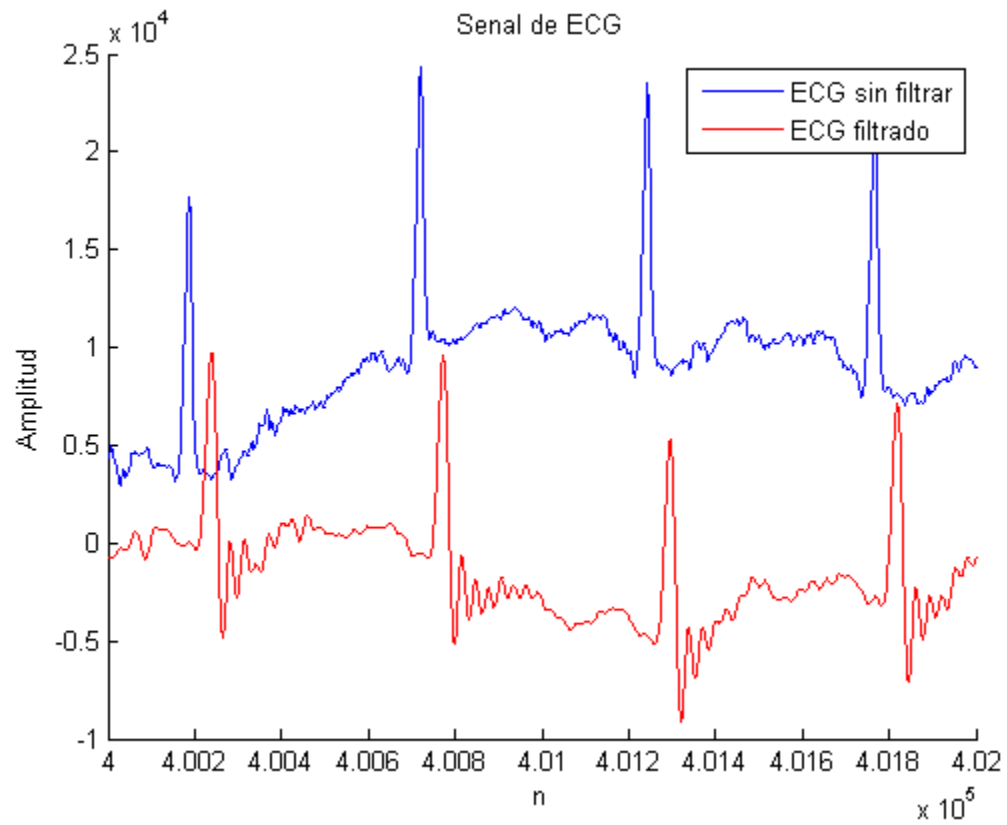
A continuación se muestran los resultados al aplicar los filtros generados a la señal de ECG3 expuesta anteriormente obteniendo los siguientes resultados:

Primer filtrado: Filtro Butterworth



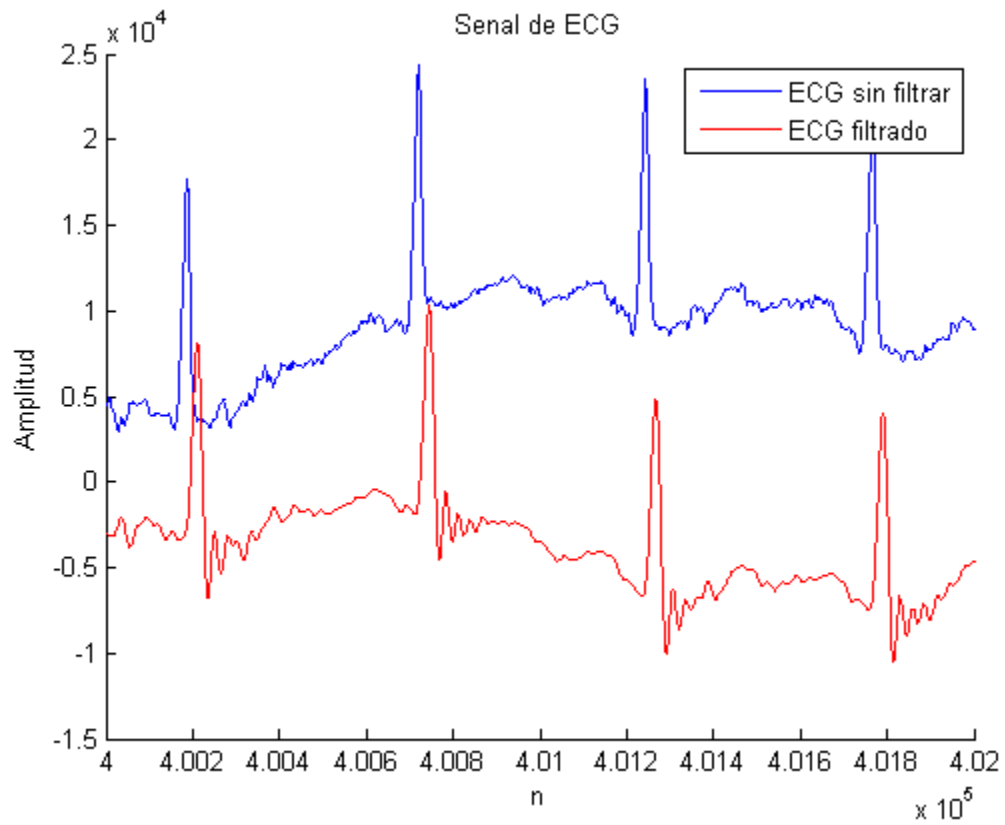
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 1



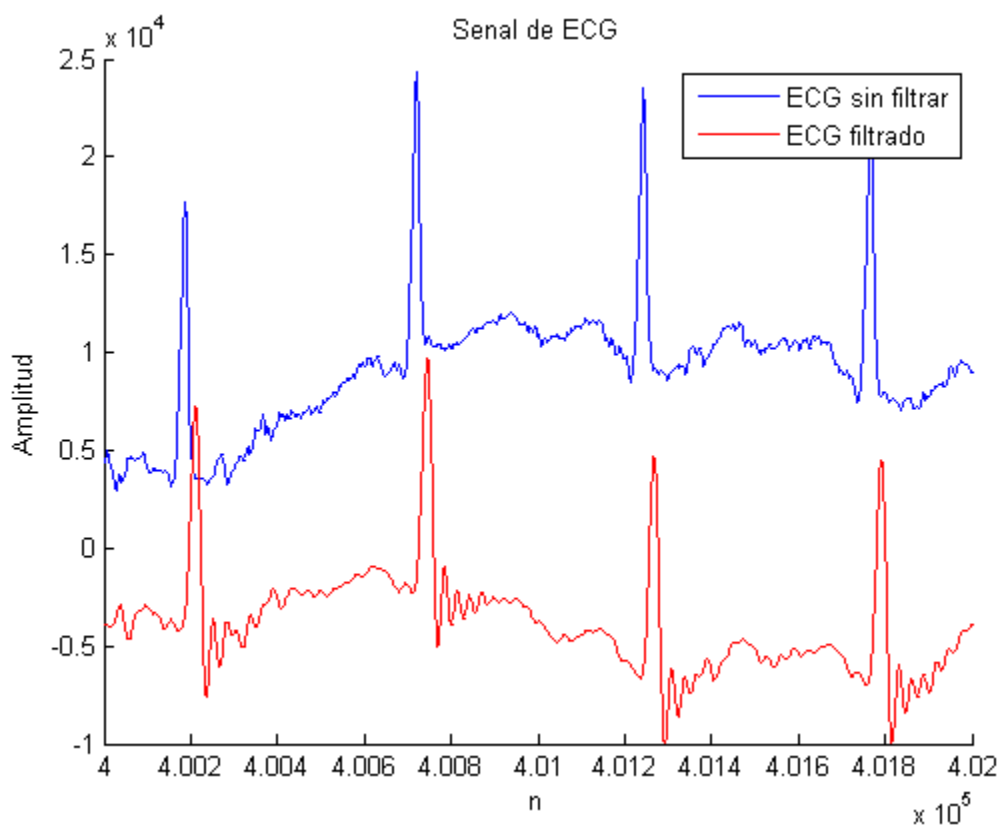
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 2



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

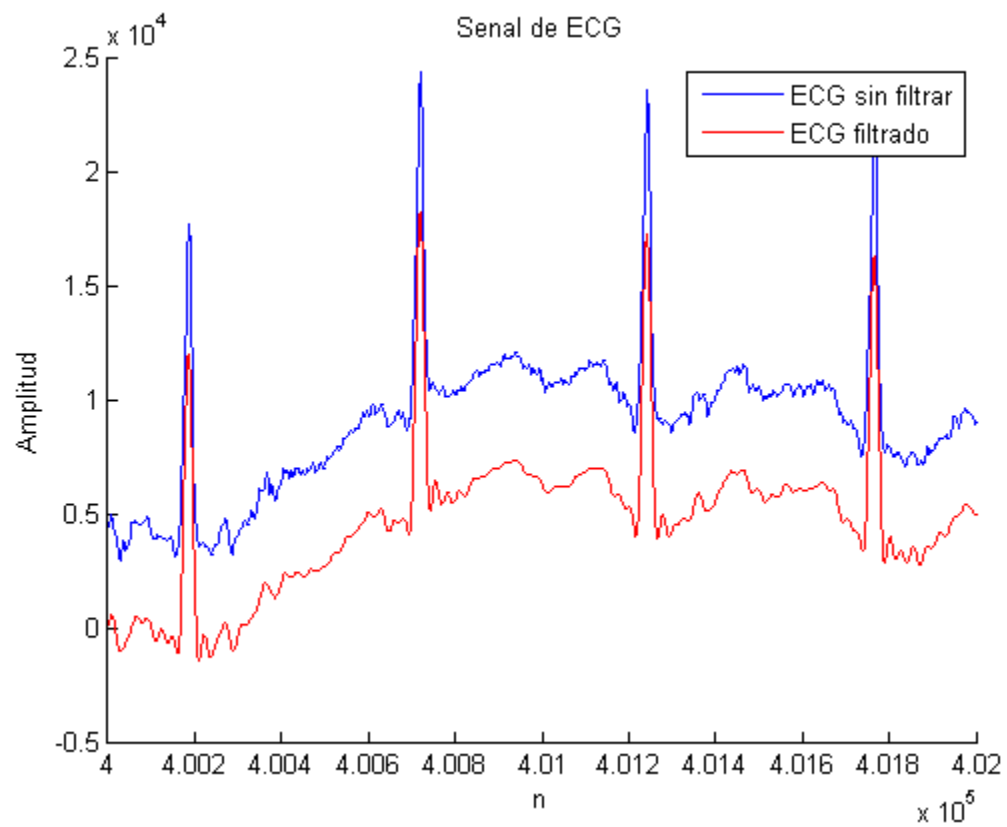
#### Cuarto filtrado: Filtro Elíptico



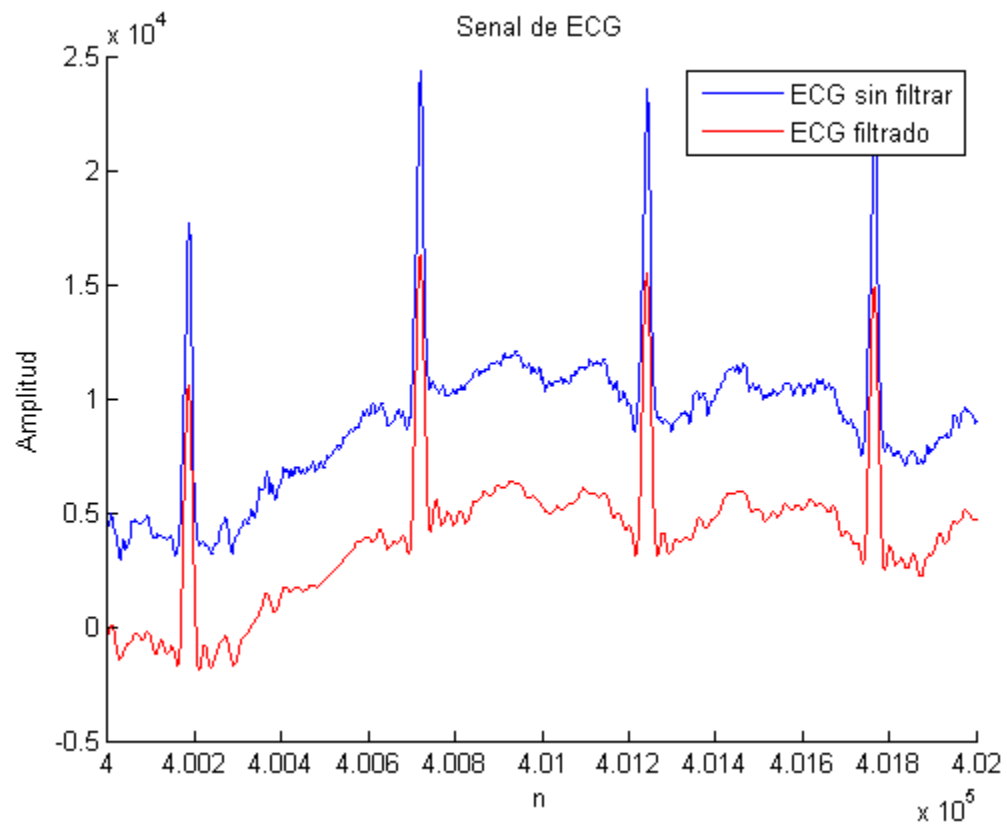
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

A continuación se aplica la técnica de doble filtrado mencionada cuando se analizó el desfase en la señal ECG4.

Primer filtrado: Butterworth (doble filtrado)

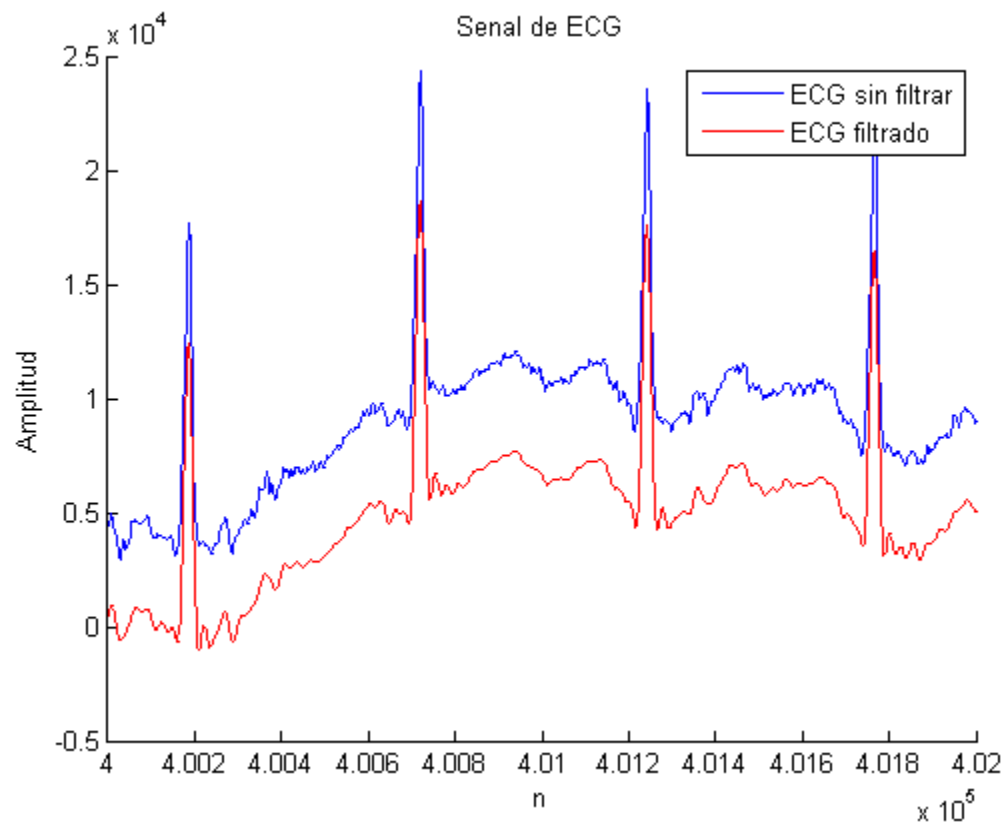


Segundo filtrado: Chebyshev 1 (doble filtrado)

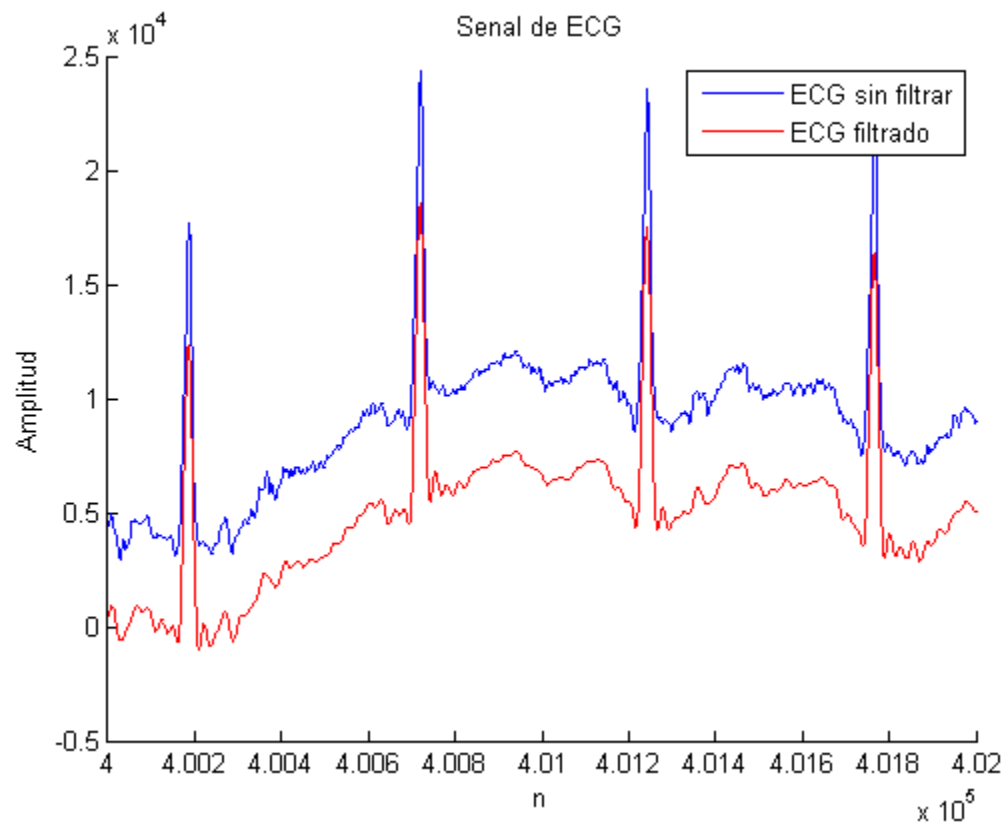




Tercer filtrado: Chebyshev 2 (doble filtrado)



Cuarto filtrado: Elíptico (doble filtrado)



A continuación realizaremos un segundo estudio igual al ya realizado para la señal ECG1 utilizando el mismo tipo de filtro sin corrección de fase.

```
EnergiaTotal = sum(abs(fft(ECG1)));
EnergiaConProc = sum(abs(fft(ECG1_filter)));
EnergiaPerdida = EnergiaTotal - EnergiaConProc;
EnergiaPerdida = 100 * (EnergiaPerdida/EnergiaTotal);
```

Energía total del ECG	4.7297e+11 J
Energía total del ECG luego de procesar	3.7706e+11 J
Energía perdida al procesar la señal	9.5902e+10 J
Porcentaje de energía perdido	% 20.2767

De los valores obtenidos se puede observar para la señal ECG4 que luego de filtrar la misma se obtuvo una perdida de energia de aproximadamente el %20 de la energía total de la señal de ECG. A continuación nuevamente se busca minimizar dicha pérdida de energía.

Se generó un filtro pasa banda con las mismas frecuencias de corte utilizadas para el tercer estudio sobre la señal ECG1

```
Fstop1 = 0.001;      % First Stopband Frequency
Fpass1 = 0.01;       % First Passband Frequency
Fpass2 = 100;        % Second Passband Frequency
Fstop2 = 120;        % Second Stopband Frequency
Astop1 = 60;         % First Stopband Attenuation (dB)
Apass  = 1;          % Passband Ripple (dB)
Astop2 = 80;         % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Porcentaje de energía perdido	% 2.9893
-------------------------------	----------

Con este filtro podemos observar que la energía perdida del filtrado es de aproximadamente %3 respecto de la energía de la señal de entrada al filtro.

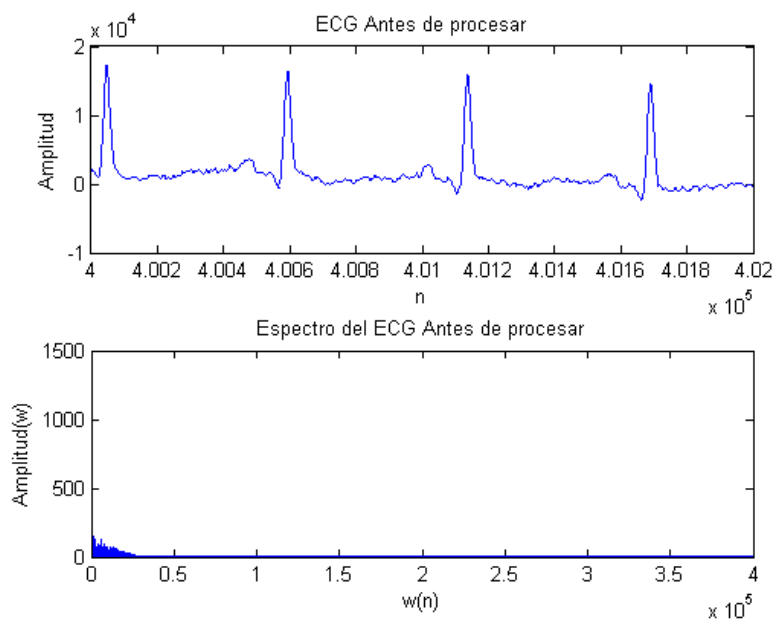
## Estudio para la señal: ECG5

A continuación se realiza el mismo estudio realizado para la señal ECG2 con la señal ECG5.

A continuación se expone la señal de ECG con la que se trabajara en este segundo estudio.

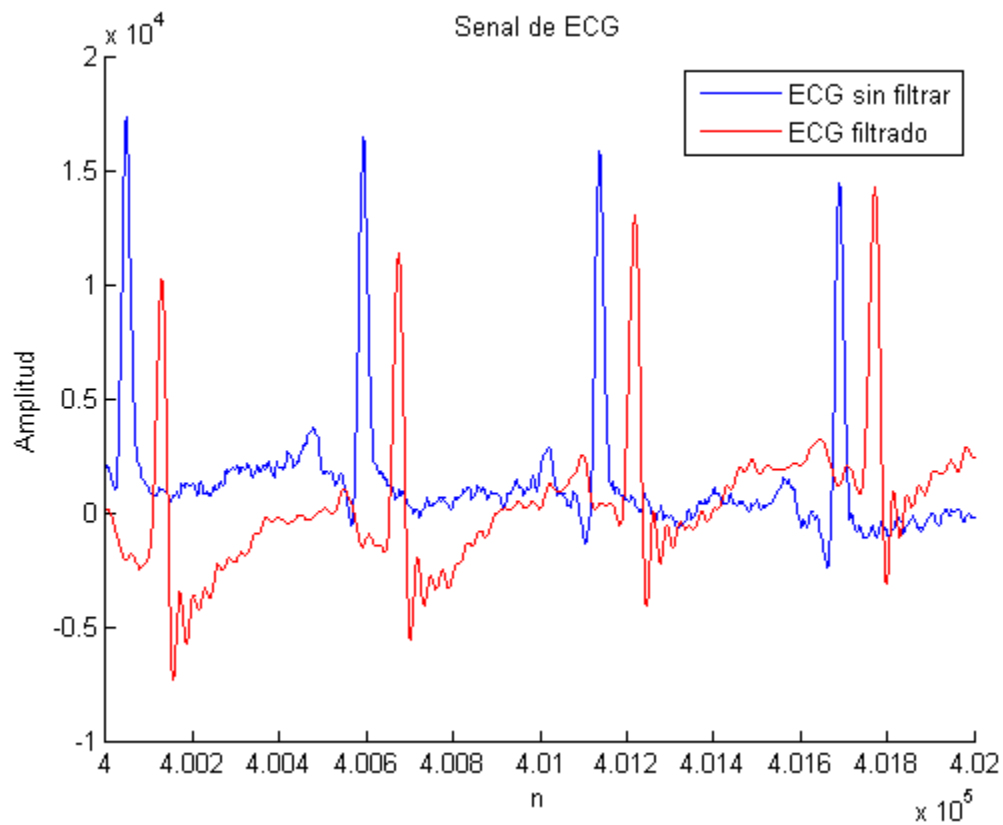
```
% Ploteo de señal de ECG - Tiempo
subplot(2,1,1); plot(4e5:4.02e5,ECG1(4e5:4.02e5));
xlabel('n'), ylabel('Amplitud');
title('ECG Antes de procesar');
% Ploteo de señal ECG - Frecuencia

subplot(2,1,2);
ECG1_tf = 2 / length(ECG1) * fft(ECG1);
plot(abs(ECG1_tf(1:length(ECG1_tf)/2)));
xlabel('w(n)'), ylabel('Amplitud(w)');
title('Espectro del ECG Antes de procesar');
```



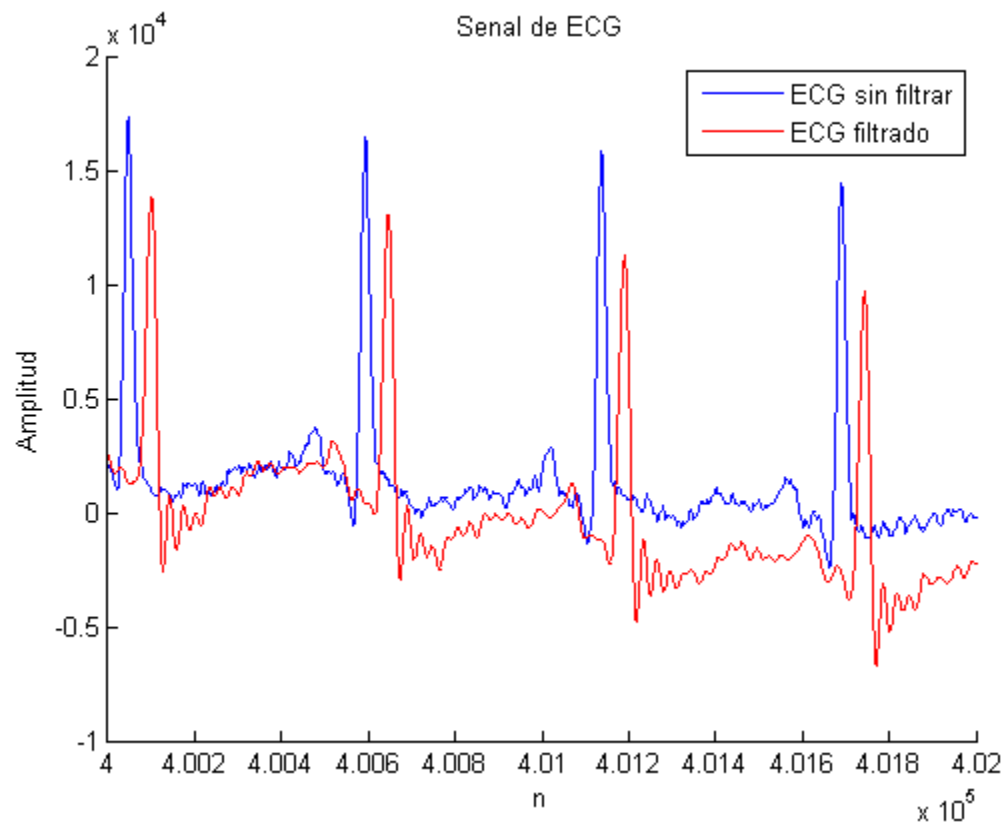
A continuación se muestran los resultados al aplicar los filtros generados a la señal de ECG5 expuesta anteriormente obteniendo los siguientes resultados:

Primer filtrado: Filtro Butterworth



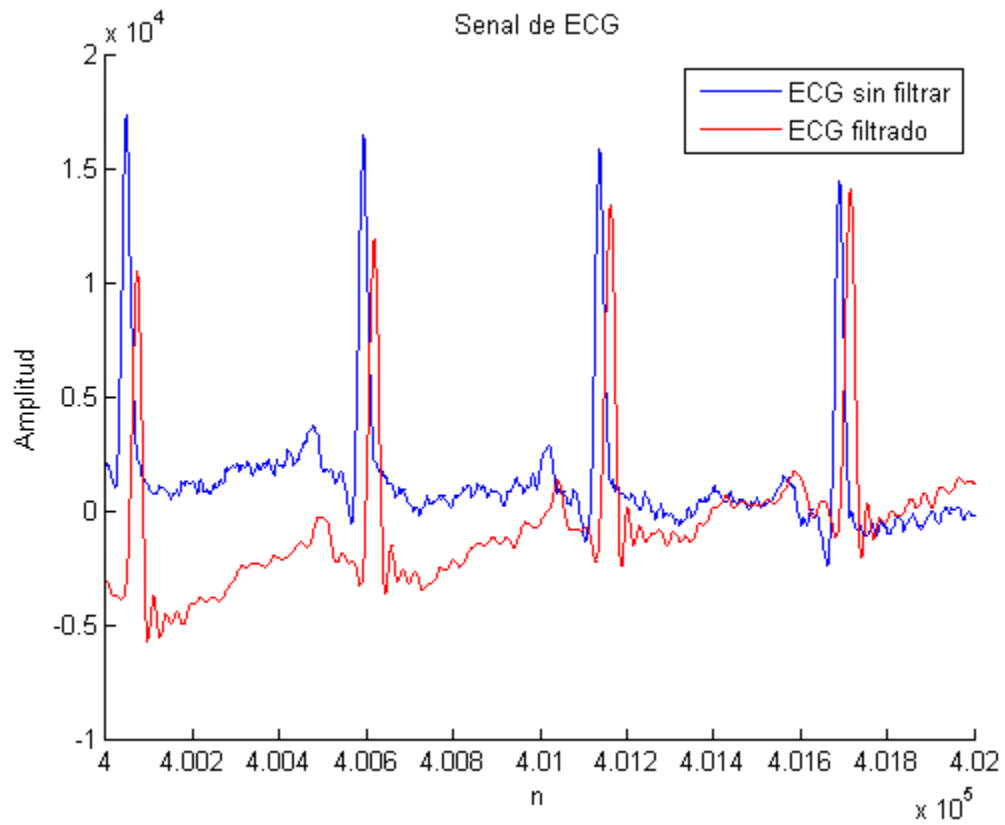
Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

Segundo filtrado: Filtro Chebyshev 1



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

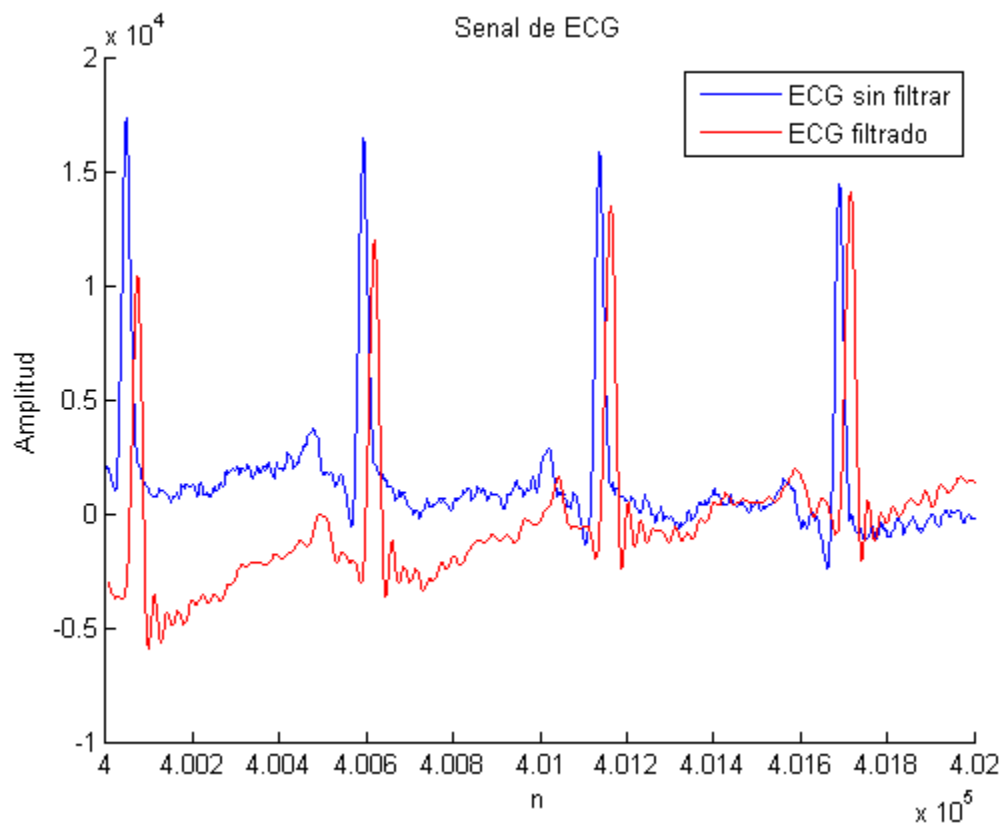
Segundo filtrado: Filtro Chebyshev 2



Puede observarse una un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.



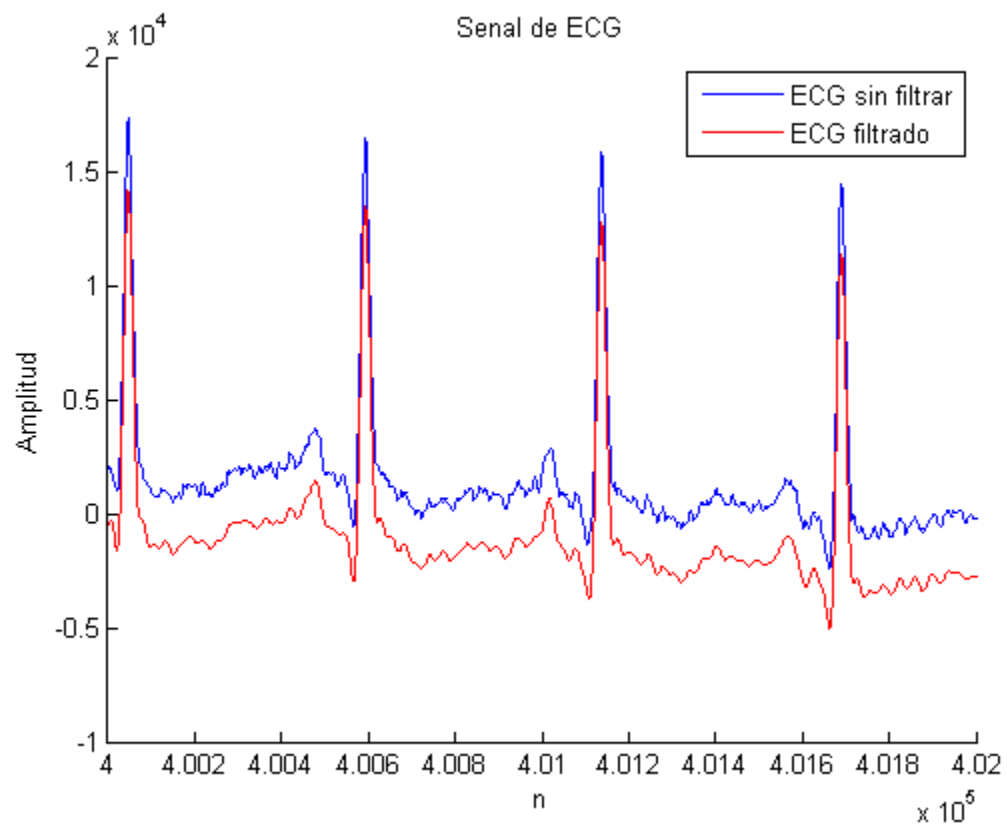
#### Cuarto filtrado: Filtro Elíptico



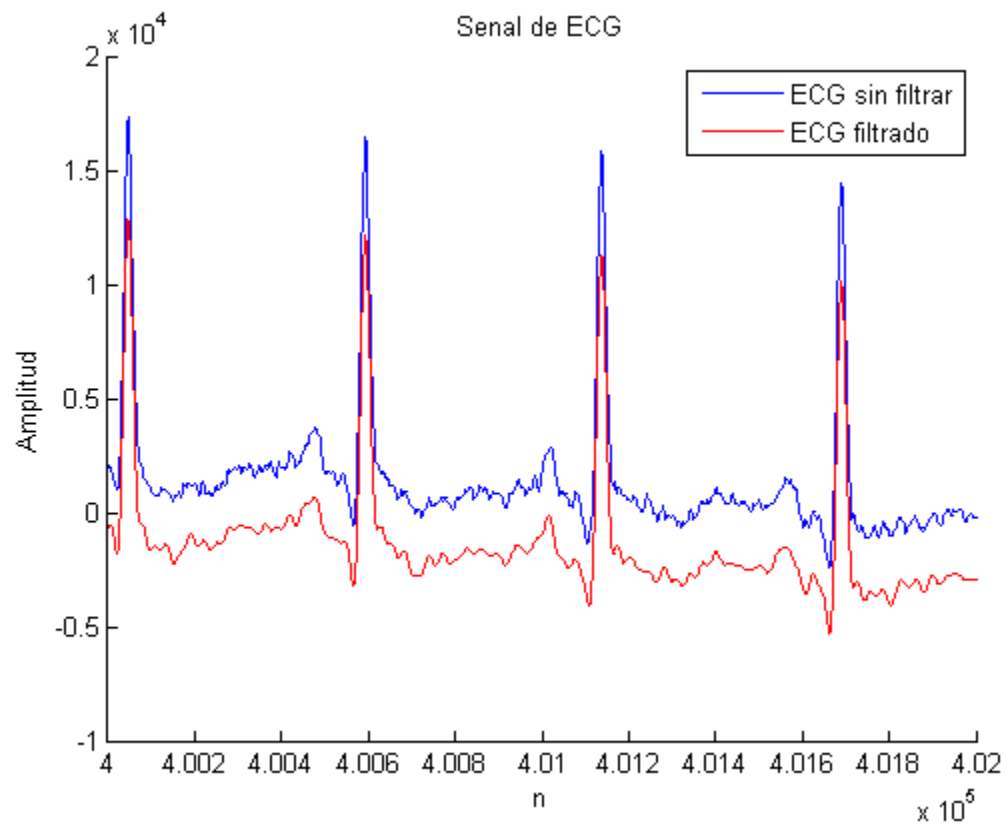
Puede observarse un desfase entre la señal de ECG y la señal filtrada con el filtro generado, al finalizar con este primer estudio se dará una solución a este desfase de forma tal de mitigar dicho problema.

A continuación se aplica la técnica de doble filtrado mencionada cuando se analizó el desfase en la señal ECG5.

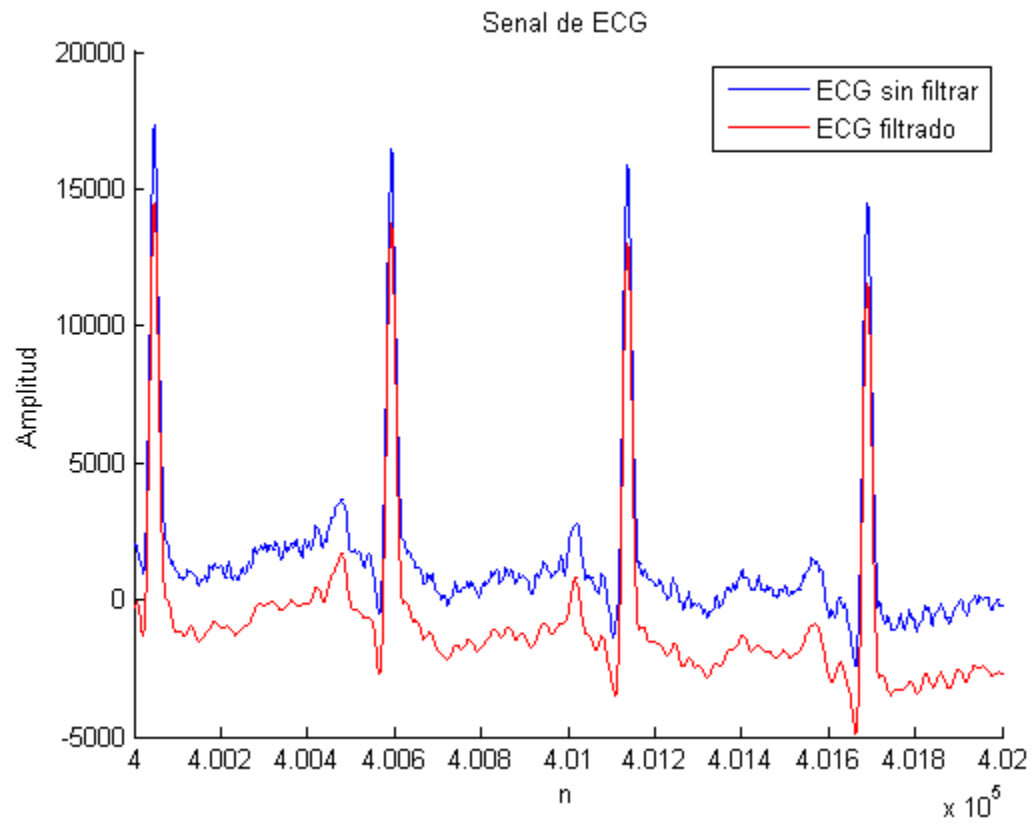
Primer filtrado: Butterworth (doble filtrado)



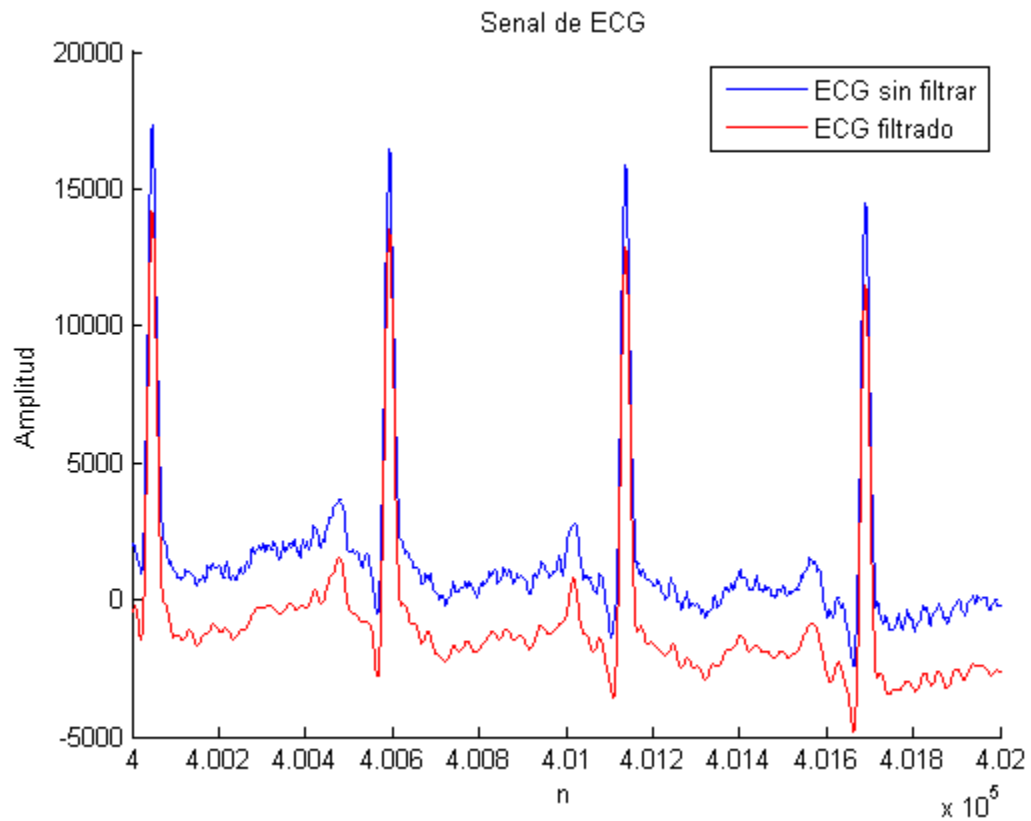
Segundo filtrado: Chebyshev 1 (doble filtrado)



Tercer filtrado: Chebyshev 2 (doble filtrado)



Cuarto filtrado: Elíptico (doble filtrado)



Se puede observar en los gráficos expuestos que en los cuatro filtrados realizados sobre la señal de ECG la señal a la salida del sistema se obtuvo en fase con la señal de entrada al filtro motivo por el cual se puede concluir en que el método mencionado y luego utilizado fue efectivo al igual que cuando se trabajó con la señal ECG1.

A continuación realizaremos un segundo estudio igual al ya realizado para la señal ECG1 utilizando el mismo tipo de filtro sin corrección de fase.

```
EnergiaTotal = sum(abs(fft(ECG1)));  
EnergiaConProc = sum(abs(fft(ECG1_filter)));  
EnergiaPerdida = EnergiaTotal - EnergiaConProc;  
EnergiaPerdida = 100 * (EnergiaPerdida/EnergiaTotal);
```

Energía total del ECG	4.7789e+11 J
Energía total del ECG luego de procesar	4.1261e+11 J
Energía perdida al procesar la señal	6.5278e+10 J
Porcentaje de energía perdido	% 13.6596

De los valores obtenidos se puede observar para la señal ECG5 que luego de filtrar la misma se obtuvo una perdida de energia de aproximadamente el %13.6 de la energía total de la señal de ECG. A continuación nuevamente se busca minimizar dicha pérdida de energía.

Se generó un filtro pasa banda con las mismas frecuencias de corte utilizadas para el tercer estudio sobre la señal ECG1

```
Fstop1 = 0.001;      % First Stopband Frequency
Fpass1 = 0.01;       % First Passband Frequency
Fpass2 = 100;        % Second Passband Frequency
Fstop2 = 120;        % Second Stopband Frequency
Astop1 = 60;         % First Stopband Attenuation (dB)
Apass  = 1;          % Passband Ripple (dB)
Astop2 = 80;         % Second Stopband Attenuation (dB)
match  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
                    Astop2, Fs);
Butterworth = design(h, 'butter', 'MatchExactly', match);
```

Porcentaje de energía perdido	% 2.8896
-------------------------------	----------

Con este filtro podemos observar que la energía perdida del filtrado es de aproximadamente %3 respecto de la energía de la señal de entrada al filtro.

## TRABAJO PRÁCTICO 5 - EJERCICIO 3

Implemente el siguiente filtro:

$$y[n] = \underset{i=n-w}{\overset{n+w}{\text{med}}} x[i],$$

Siendo

$$\text{med } x[i] = \arg \min_{\mu} \sum_{i=1}^N |x[i] - \mu|,$$

Este filtro se conoce como filtro de mediana. Es el equivalente robusto del filtro de media móvil en el TP3. Aplique dicho filtro en una ventana de 200mS a las señales de ECG (fs = 1000Hz) y compare respecto al resultado de un filtro de media móvil que use la misma ventana.

Ahora modifique la función mediana por min y max. Podría generalizar este filtro para un percentil arbitrario, sabiendo que min, med y max son los percentiles 0,50 y 100% usando la función pretile.



Para este ejercicio se desarrolló el algoritmo: Mediana, Media y Filtro de mediana los cuales pueden encontrarse al final del ejercicio en este informe o en la carpeta: funciones básicas con los nombres: “mediana”, “FilterMedia” y “FilterMedian”.

Se utilizó el filtro de mediana a una ventana de 200mS a la señal de ECG1, cuya frecuencia de muestreo fue de 1000Hz luego se comparó al mismo con el robusto filtro de media móvil visto en el trabajo práctico 3 de la materia.

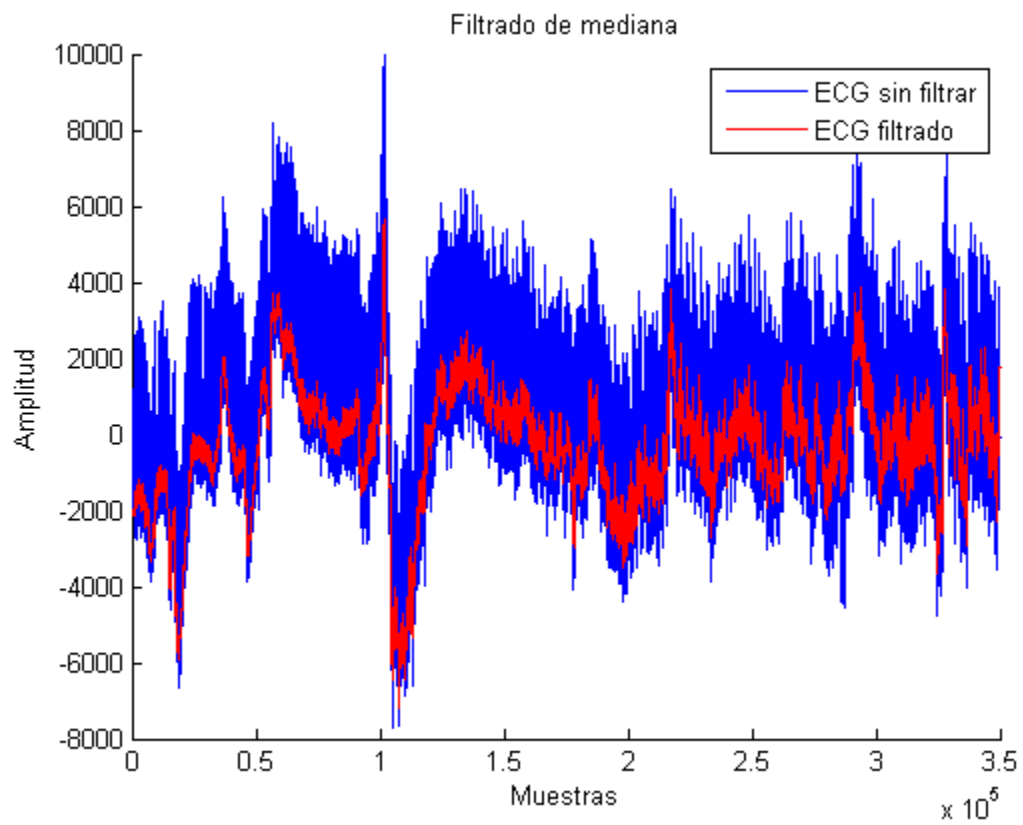
Ventana	200 muestras
Frecuencia de sampleo	1000Hz

Como primer paso de estudio se filtró la señal de ECG con el filtro de Media móvil obteniendo los siguientes resultados:

```
tic; EcgFilterMedia = FilterMedia(ECG1,LengW); MediaTime = toc;
```

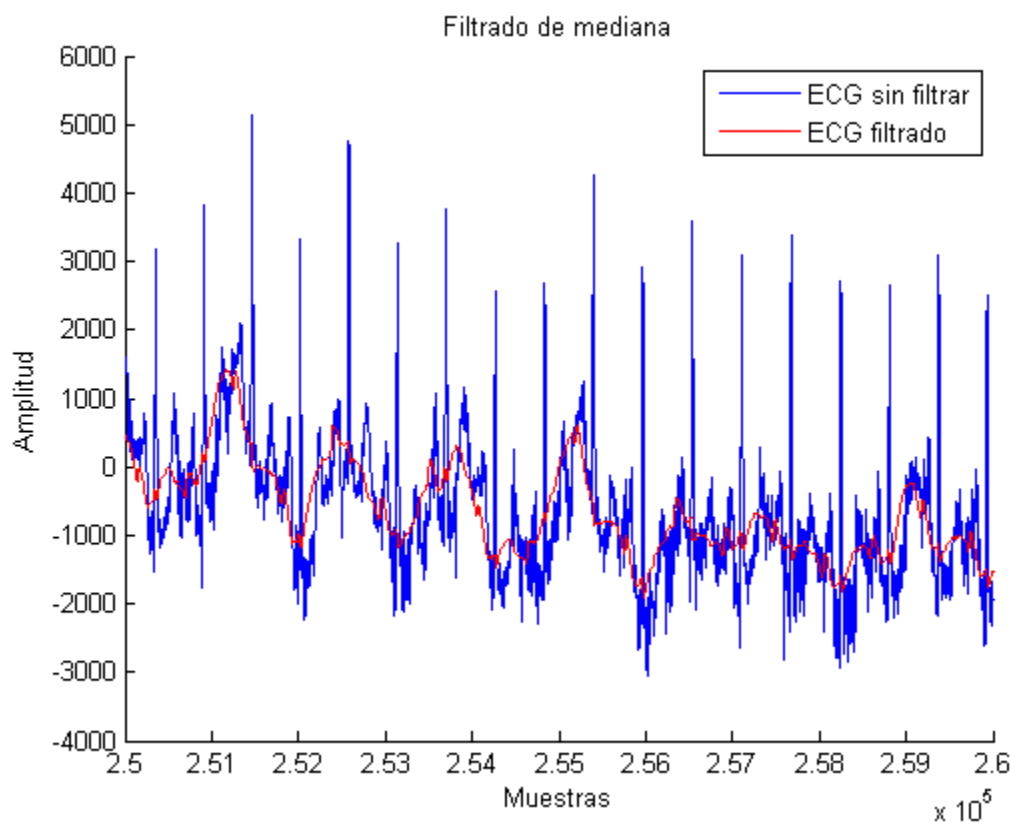
Tiempo de filtrado con filtro de media	5.7530 Segundos
--	-----------------

A partir del filtrado con el filtro de media móvil se obtuvo la siguiente señal de salida:



*Nota: El título del gráfico es erróneo, este gráfico es el resultado de filtrar con un filtro robusto de media móvil.*

Aplicándole un zoom a la señal se puede observar tanto la señal de entrada al filtro como la salida o señal filtrada de forma más clara.



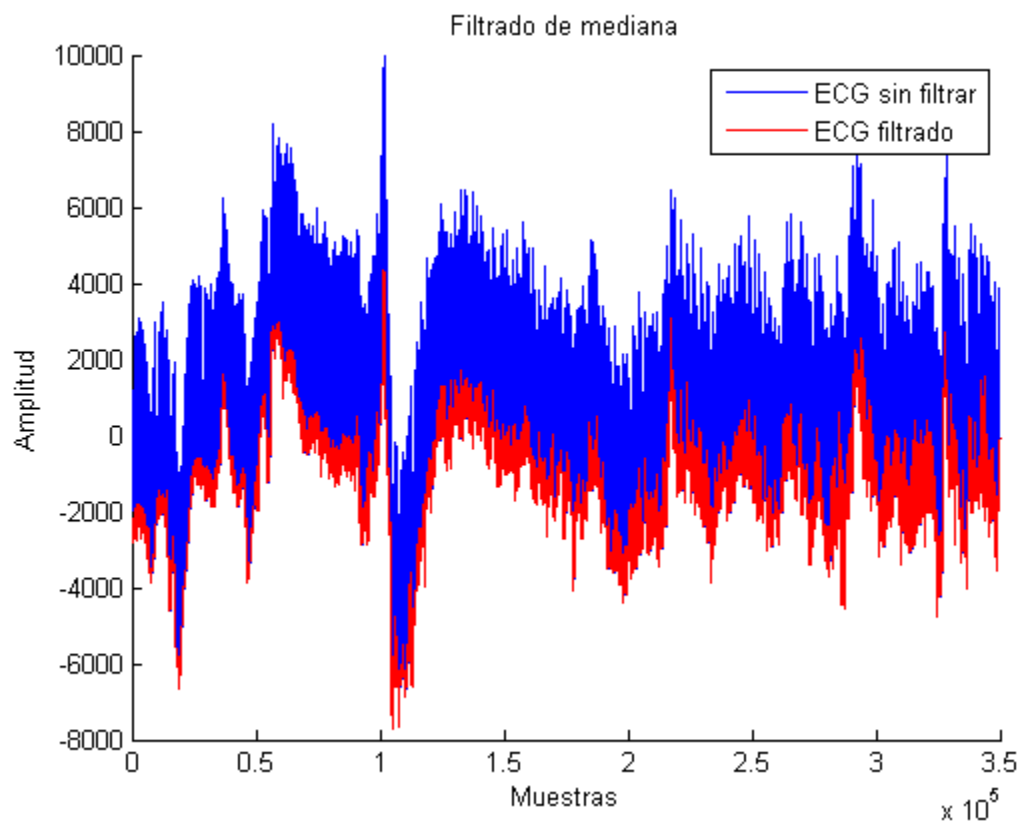
*Nota: El título del gráfico es erróneo, este gráfico es el resultado de filtrar con un filtro robusto de media móvil.*

Como segundo paso de estudio se filtró la señal de ECG con el filtro de Mediana – percentil 0, obteniendo los siguientes resultados:

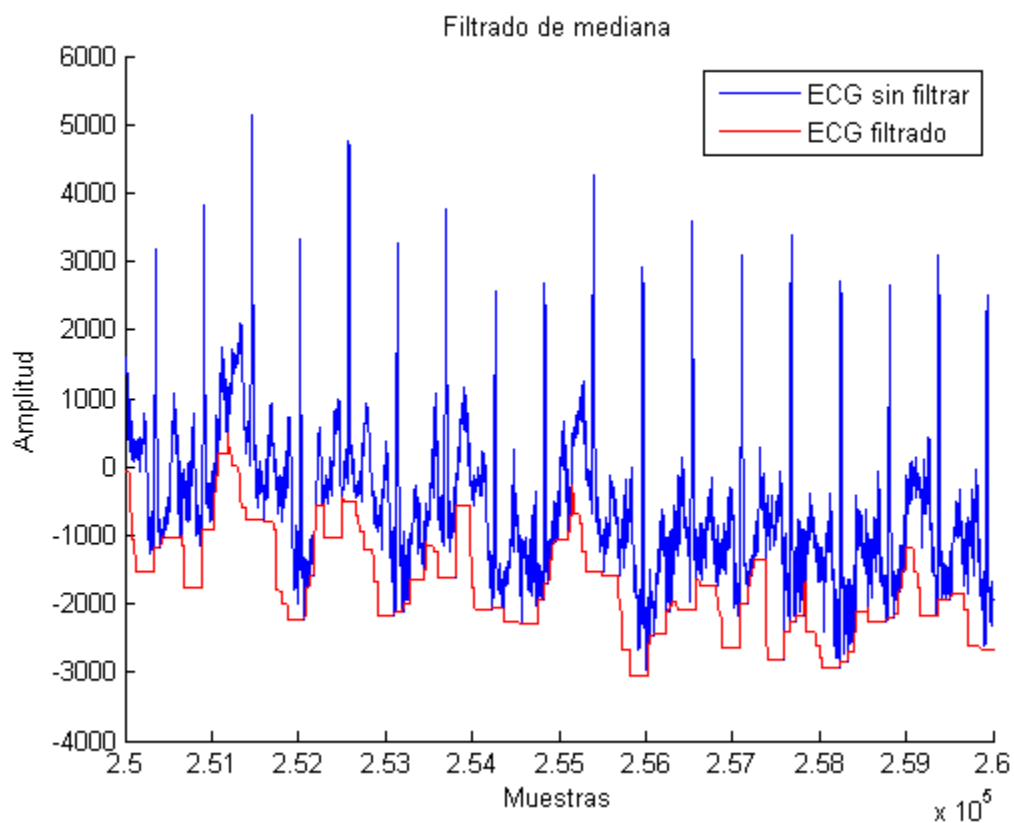
```
percentil = 0;  
tic; EcgFilterMedian0 = FilterMedian(ECG1,Lengw,percentil); MedianTime =toc;
```

Tiempo de filtrado con filtro de mediana percentil 0	58.8607 Segundos
--	---------------------

A partir del filtrado con el filtro de Mediana percentil 0 se obtuvo la siguiente señal de salida:



Aplicándole un zoom a la señal se puede observar tanto la señal de entrada al filtro como la salida o señal filtrada de forma más clara.



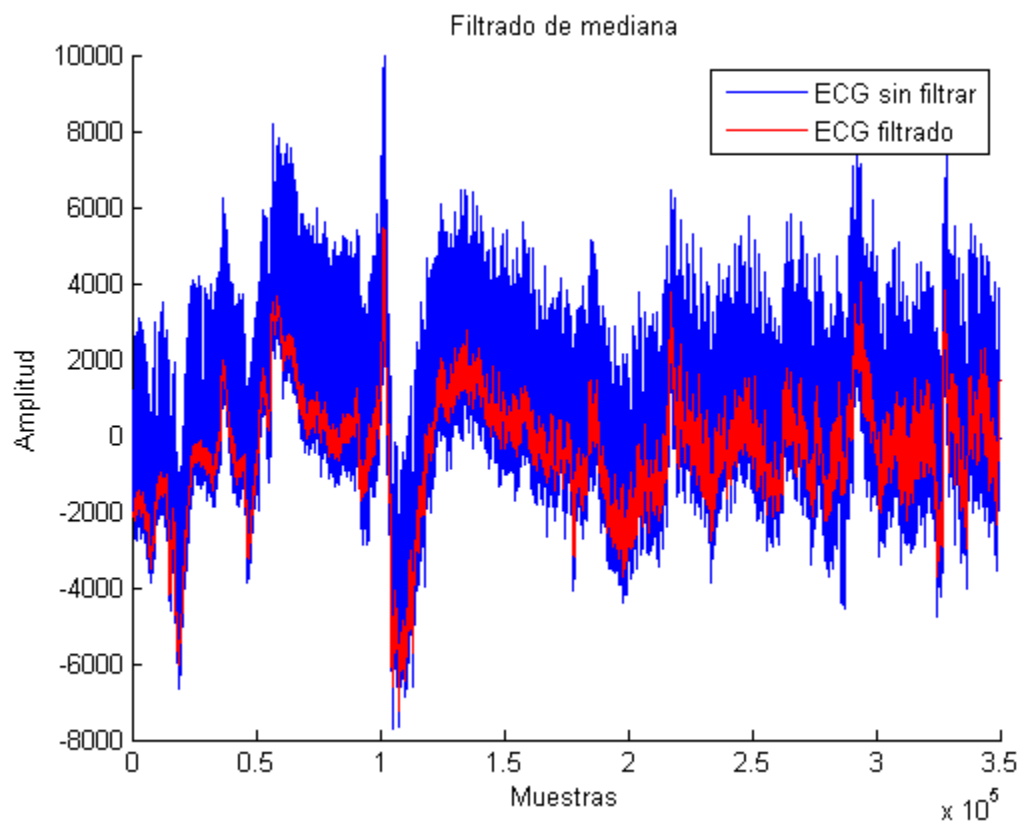
Se puede observar que cuando se selecciona para el filtro de mediana un percentil 0 la señal de ECG filtrada es una señal que “sigue” los mínimos de la señal de ECG sin filtrar de a tramos equivalentes al tamaño de la ventana.

Como tercer paso de estudio se filtró la señal de ECG con el filtro de Mediana – percentil 50, obteniendo los siguientes resultados:

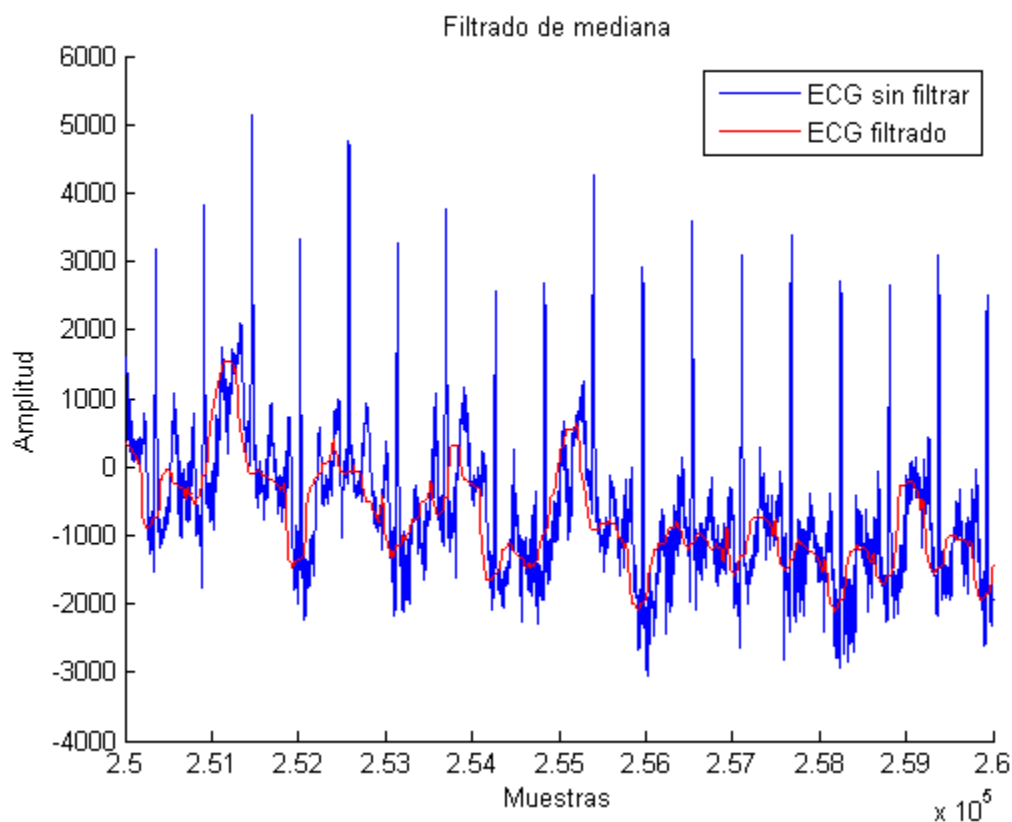
```
percentil = 50;
tic; EcgFilterMedian50 = FilterMedian(ECG1,Lengw,percentil); MedianTime =toc;
```

Tiempo de filtrado con filtro de mediana percentil 50	6.4092 Segundos
---	--------------------

A partir del filtrado con el filtro de Mediana percentil 50 se obtuvo la siguiente señal de salida:



Aplicándole un zoom a la señal se puede observar tanto la señal de entrada al filtro como la salida o señal filtrada de forma más clara.



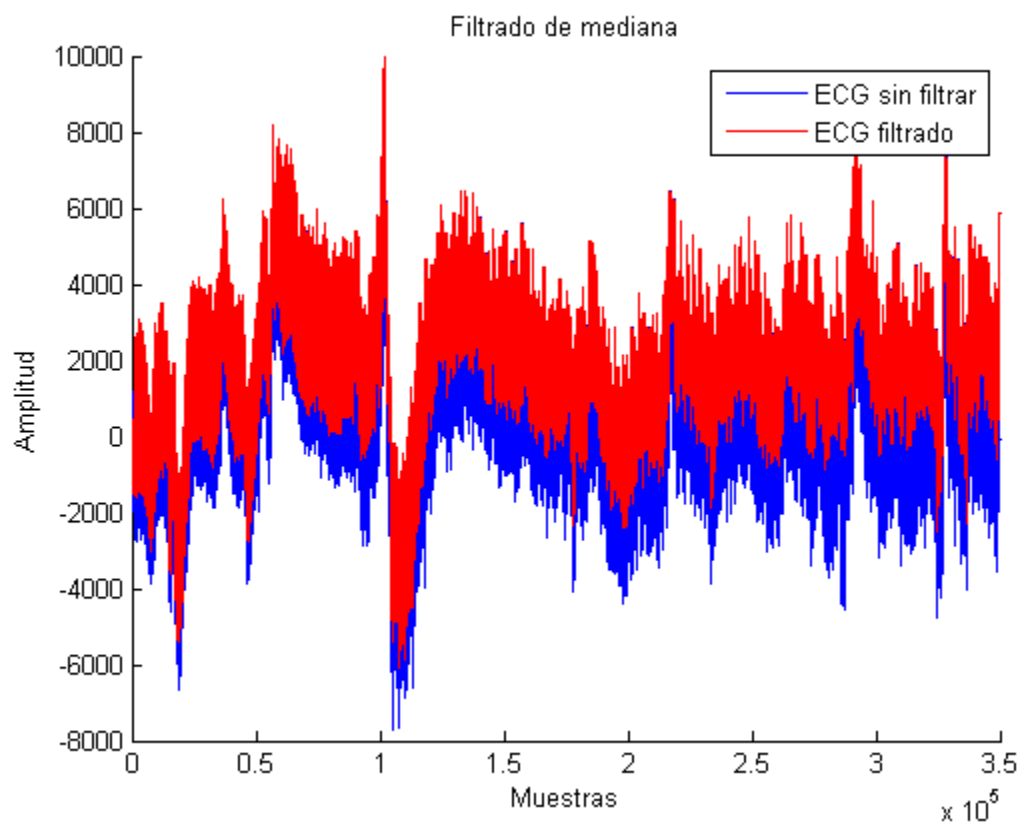
Se puede observar que cuando se selecciona para el filtro de mediana un percentil 50 la señal de ECG filtrada es una señal que “sigue” los valores medios de la señal de ECG sin filtrar de a tramos equivalentes al tamaño de la ventana.

Como tercer paso de estudio se filtró la señal de ECG con el filtro de Mediana – percentil 100, obteniendo los siguientes resultados:

```
percentil = 100;  
tic; EcgFilterMedian100 = FilterMedian(ECG1,Lengw,percentil); MedianTime =toc;
```

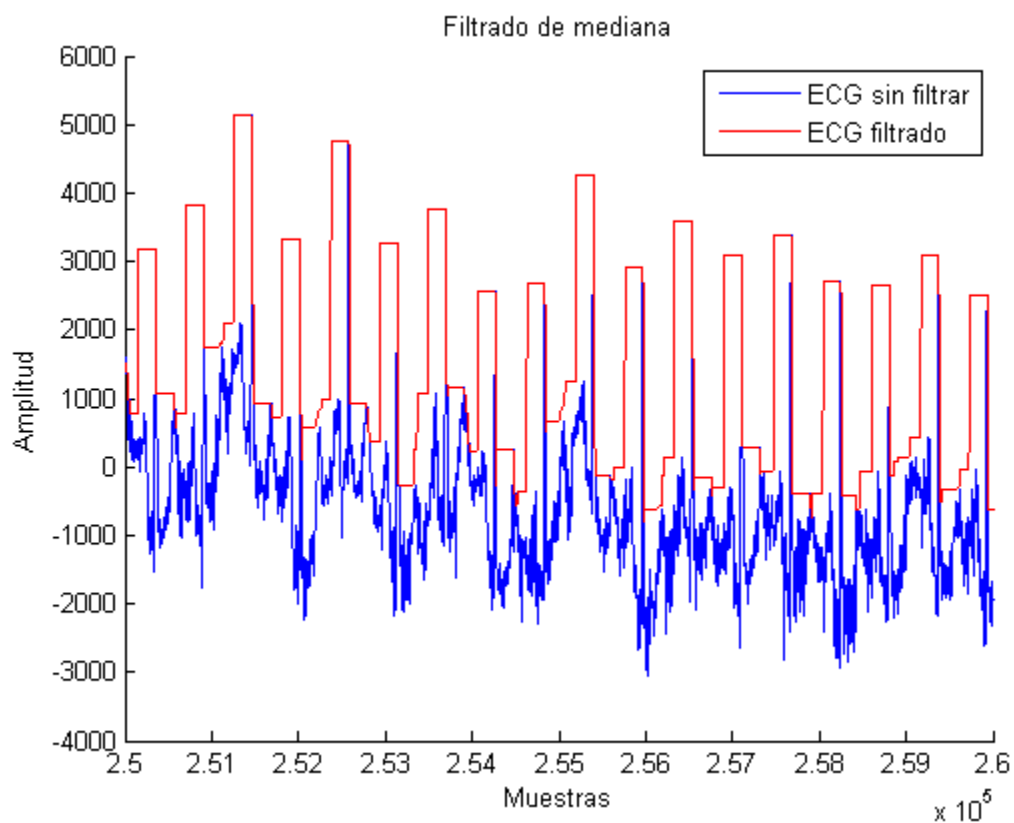
Tiempo de filtrado con filtro de mediana percentil 100	6.1104 Segundos
--	-----------------

A partir del filtrado con el filtro de Mediana percentil 100 se obtuvo la siguiente señal de salida:





Aplicándole un zoom a la señal se puede observar tanto la señal de entrada al filtro como la salida o señal filtrada de forma más clara.



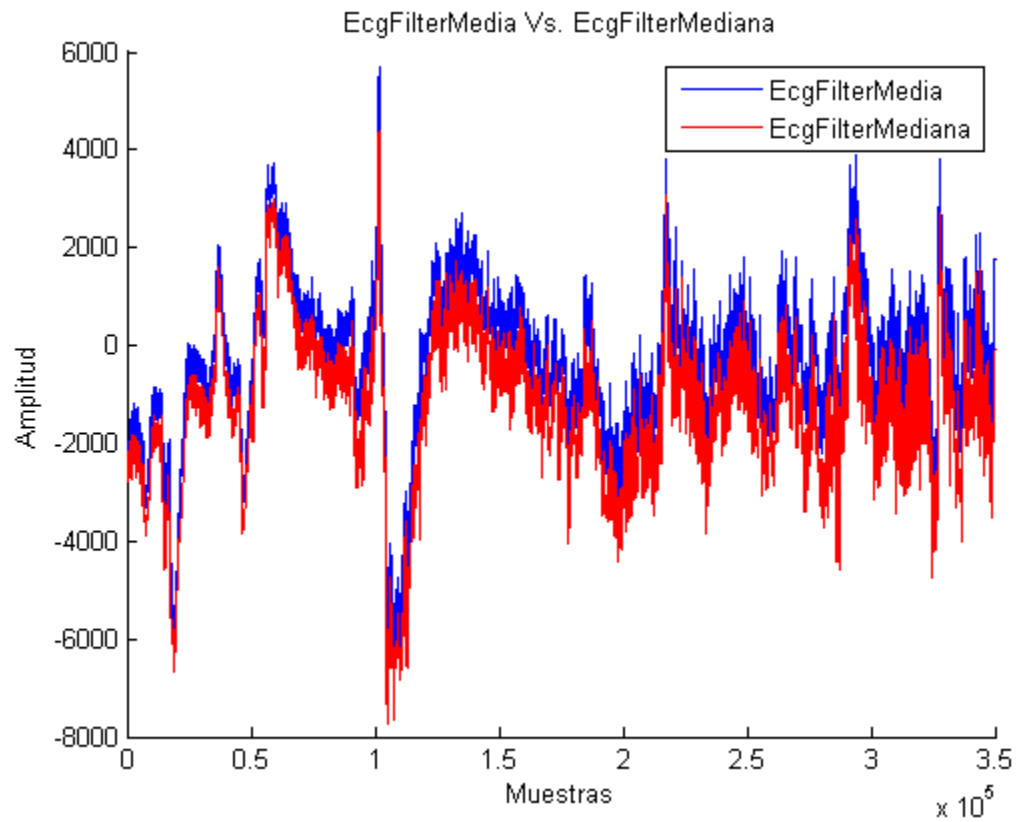
Se puede observar que cuando se selecciona para el filtro de mediana un percentil 100 la señal de ECG filtrada es una señal que “sigue” los máximos de la señal de ECG sin filtrar de a tramos equivalentes al tamaño de la ventana.

A continuación se expone una tabla comparativa de tiempos entre el filtro robusto de media móvil, el filtro de mediana percentil 0, 50 y 100.

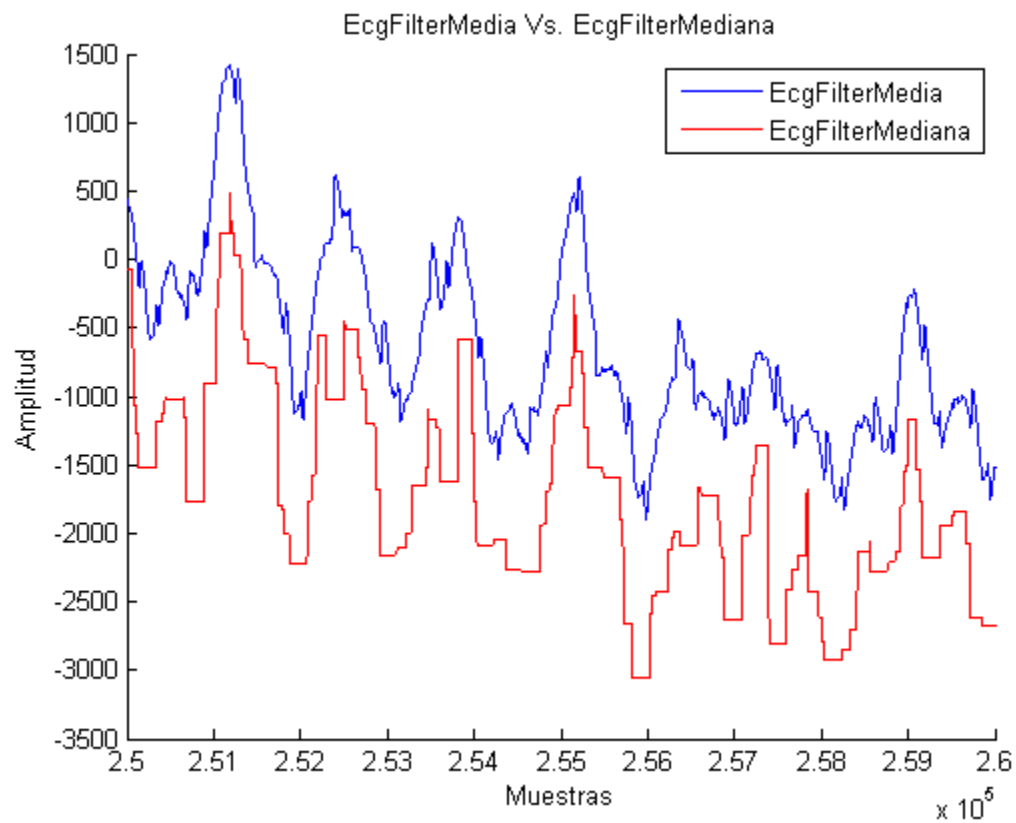
FILTRO	TIEMPO DE FILTRADO [SEGUNDOS]
FILTRO ROBUSTO DE MEDIA MÓVIL	5.7530
FILTRO DE MEDIANA PERCENTIL 0	58.8607
FILTRO DE MEDIANA PERCENTIL 50	6.4092
FILTRO DE MEDIANA PERCENTIL 100	6.1104

A continuación se exponen los resultados del filtrado de Media y Mediana para los tres percentiles

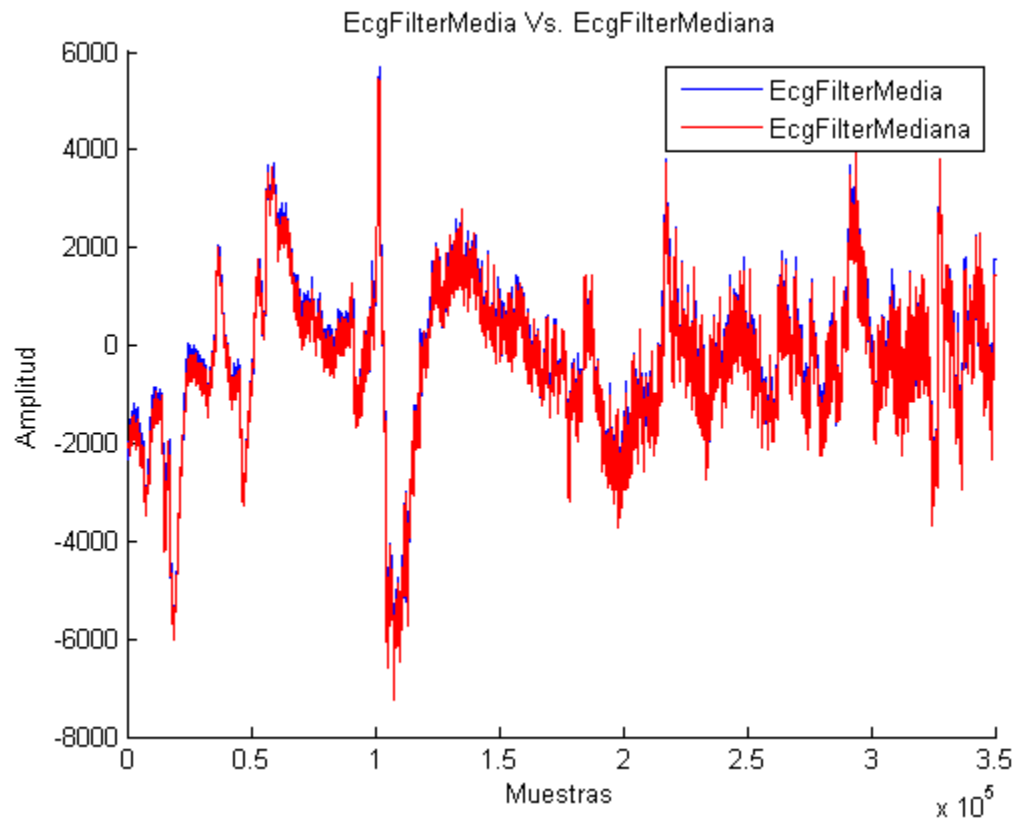
Señal filtrada con filtro de media móvil vs. Señal filtrada con filtro de mediana percentil 0



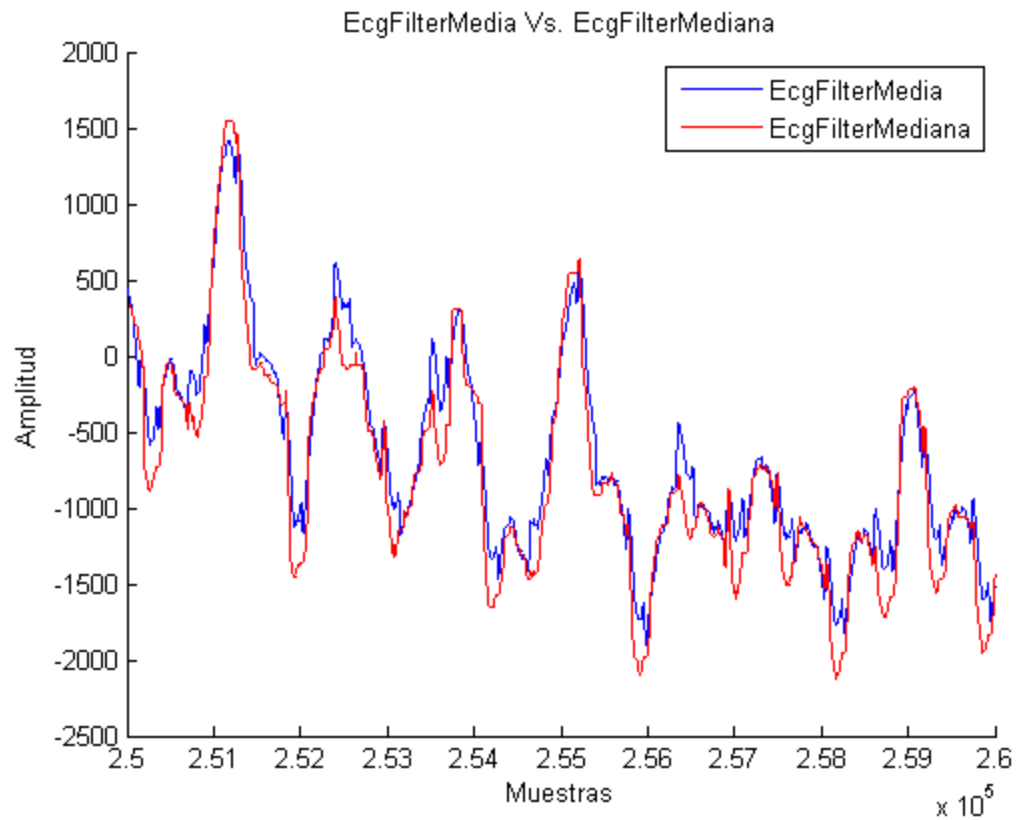
Aplicándole un zoom a la señal se puede observar tanto la señal de salida cuando se aplica un filtrado con un filtro robusto de media móvil como la señal de salida cuando se realiza un filtrado utilizando un filtro de mediana de forma más clara.



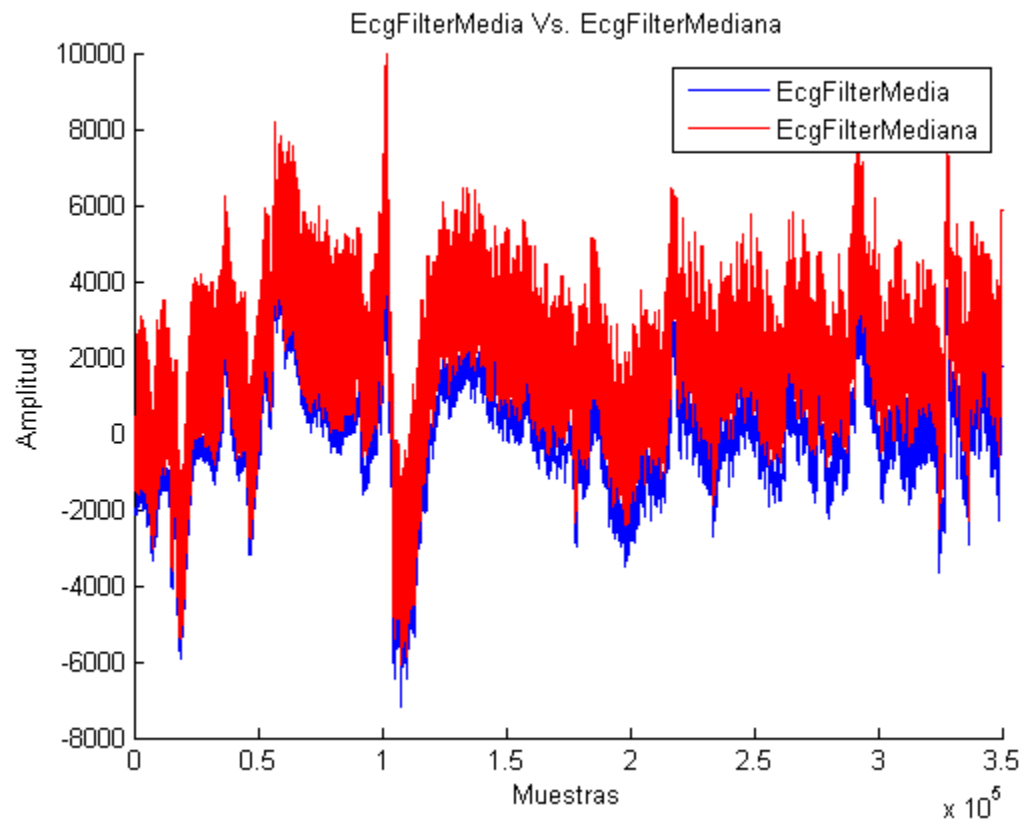
Señal filtrada con filtro de media móvil vs. Señal filtrada con filtro de mediana percentil 50



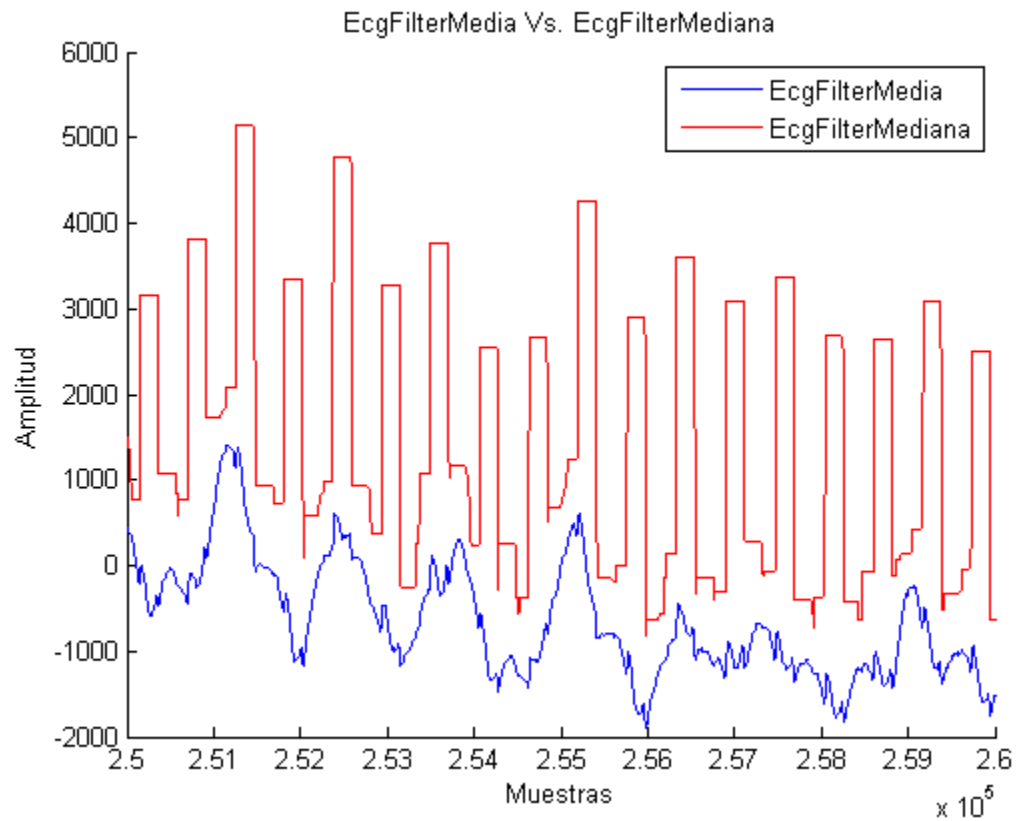
Aplicándole un zoom a la señal se puede observar tanto la señal de salida cuando se aplica un filtrado con un filtro robusto de media móvil como la señal de salida cuando se realiza un filtrado utilizando un filtro de mediana de forma más clara.



Señal filtrada con filtro de media móvil vs. Señal filtrada con filtro de mediana percentil 100



Aplicándole un zoom a la señal se puede observar tanto la señal de salida cuando se aplica un filtrado con un filtro robusto de media móvil como la señal de salida cuando se realiza un filtrado utilizando un filtro de mediana de forma más clara.





## Ejercicio 3 – Códigos

A continuación se exponen los principales códigos generados para la resolución del ejercicio

### Ejercicio 3 - Código: Filtro robusto de media móvil

```
function [out] = FilterMedia(x,Lengw)

    sizeX = length(x)/2;
    out   = zeros(1,sizeX);

    if sizeX < Lengw
        Lengw = sizeX;
    end

    for k = 1 : length(x)/2
        out(k) = mean(x(k:(k+Lengw)));
    end
    out = out';
end
```

### Ejercicio 3 - Código: Filtro de mediana para una ventana y un percentil ingresado por parámetros

```
function [out] = FilterMedian(x,Lengw,percentil)

    sizeX = length(x)/2;
    out   = zeros(sizeX,1);

    if sizeX < Lengw
        Lengw = sizeX;
    end

    for k = 1 : length(x)/2
        out(k) = mediana(x(k:(k+Lengw)),percentil);
    end
end
```

### Ejercicio 3 - Código: Función mediana para percentiles 0, 50, 100 y otros

```
function [y] = mediana(x,percentil)

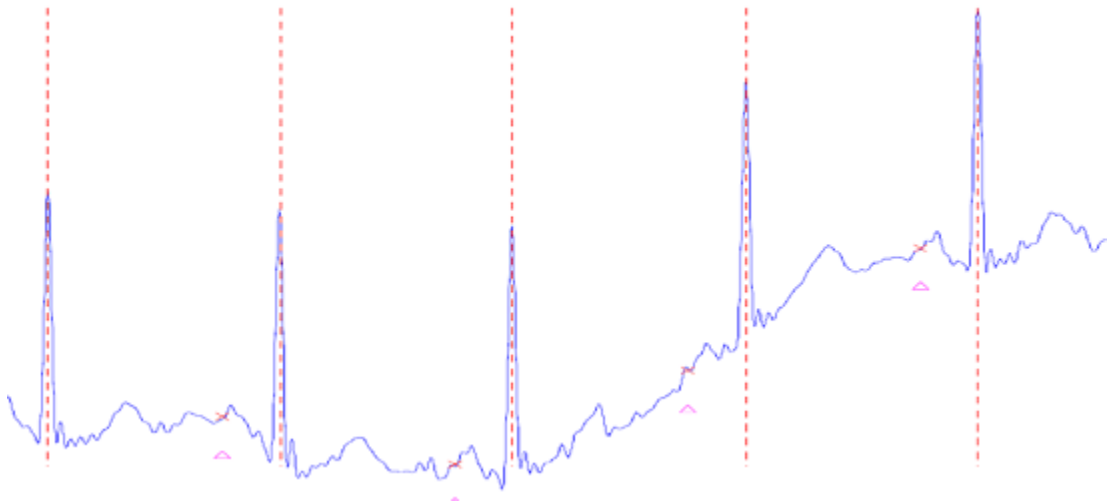
    xOrd = sort(x,'ascend');
    xOrd = round(xOrd);

    if percentil == 0
        y = xOrd(1);
    end
    if percentil == 50
        if rem(length(xOrd),2) == 0
            y = (xOrd(end/2)+xOrd(end/2+1))/2;
        else
            y = xOrd(round(end/2));
        end
    else
        if percentil == 100
            y = xOrd(end);
        else
            y = round(prctile(xOrd,percentil));
        end
    end
end
```

## TRABAJO PRÁCTICO 5 - EJERCICIO 4

Compare las siguientes estrategias para suprimir los movimientos de línea base en las señales de ECG usadas anteriormente:

- a) Estimación de la línea de base:  $V(n) = \text{med}_{600}(\text{med}_{200}x(n))$   
Siendo  $x(n)$  la señal de ECG y  $v(n)$  la estimación de la señal de la línea de base.
- a) Generar una señal de movimiento de línea de base mediante la estimación del valor del segmento PQ (el segmento que va desde la onda P hasta la onda Q), según se muestra en la figura con una cruz roja y una flecha magenta.



Para la localización del segmento PQ de cada latido, puede valerse de la localización de cada latido, provista en la variable `QRS_locN` (N es un número entero) y representa en la figura como líneas rojas verticales. Recuerde que la señal de línea de base debe estar muestreada a la misma frecuencia que el ECG, por lo que debería remuestrear dicha señal (puede usar la función de interpolación spline para ello)

Luego para ambos casos de la señal de ECG sin movimiento de línea de base sería:

$$X_c(n) = X(n) - V(n)$$

Luego para ambas estrategias se pide discutir los resultados obtenidos visualmente, comparar los tiempos de ejecución, y discutir ventajas y desventajas de ambos enfoques.

En este ejercicio se plantean dos formas de suprimir los movimientos de línea de base en las señales de ECG. El primer método se realizara mediante la expresión:

$$V(n) = med_{600}(med_{200}x(n))$$

Siendo  $x(n)$  la señal de ECG y  $v(n)$  la estimación de la señal de línea de base.

Una vez obtenida la señal de línea de base se le restará a la señal de ECG obteniendo así la señal de ECG filtrada.

El segundo método consiste en la generación de una señal de movimiento de línea de base mediante la estimación del valor del segmento PQ, una vez obtenido dichos puntos a través de una interpolación se obtendrá la señal de línea y se procederá nuevamente a restar la señal de ECG por dicha señal obteniendo así la señal de ECG filtrada.

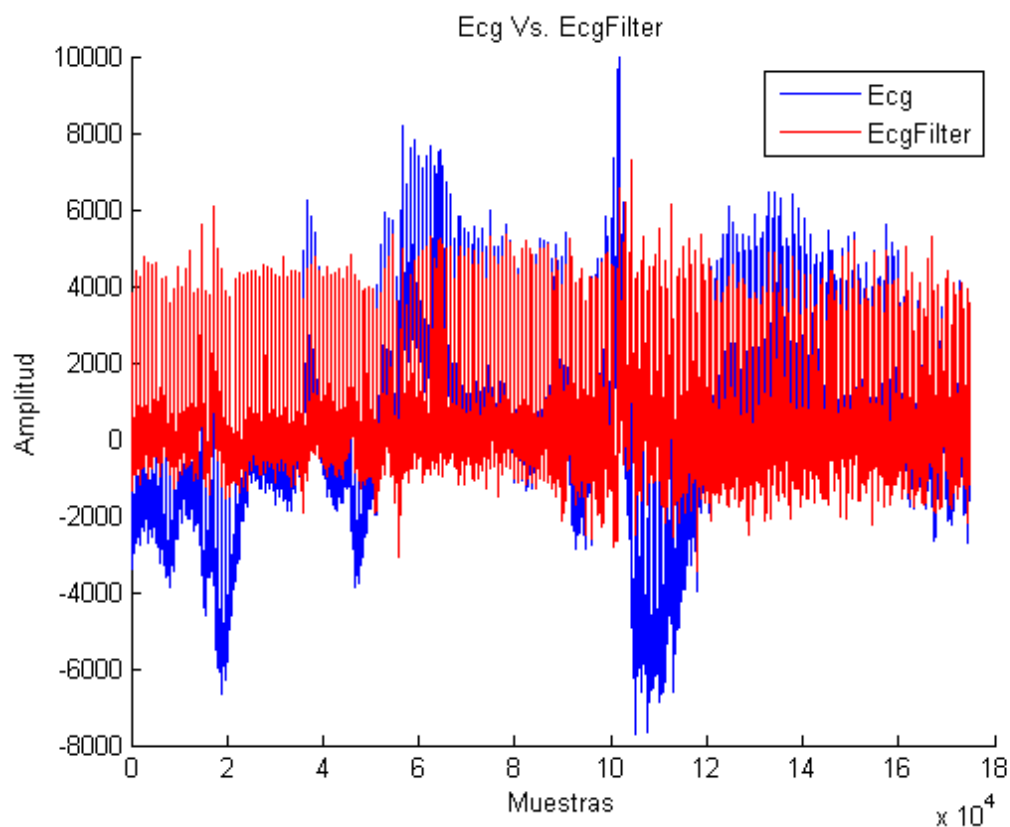
A continuación se exponen los experimentos planteados, los resultados obtenidos y las pertinentes conclusiones obtenidas luego de realizar el procesamiento sobre la señal ECG1 propuesta por la cátedra y que se adjunta junto al informe.

Primer método:  $V(n) = med_{600}(med_{200}x(n))$

Para este primer método, utilizamos la función: “mediana.m” y “filtromediana.m”, funciones implementadas en el ejercicio 3 de este trabajo práctico y que se adjuntan al informe dentro de la carpeta “funciones básicas”.

El filtrado de mediana se hizo a través de dos ventanas, siendo la primera de 200nS y la segunda de 600nS además el filtro de mediana es de percentil 50.

A continuación se muestra el resultado obtenido, para el primer método, al aplicar el filtro a la señal de ECG1.



*Nota: Se graficó una cuarta parte de la señal de ECG completa, esto es debido a la forma en la que implementé el filtro de mediana. Además se considera que de esta forma se puede observar de forma más clara ambas señales.*

Se puede observar que el filtro utilizado cumplió de forma efectiva con el propósito para el cual fue realizado.

Tiempo de filtrado	11.4735 Segundos
--------------------	------------------

A partir de aplicar el filtro:  $V(n) = \text{med}_{600}(\text{med}_{200}x(n))$  se observó que el mismo tiene una demora en el filtrado de aproximadamente 11.5 segundos.

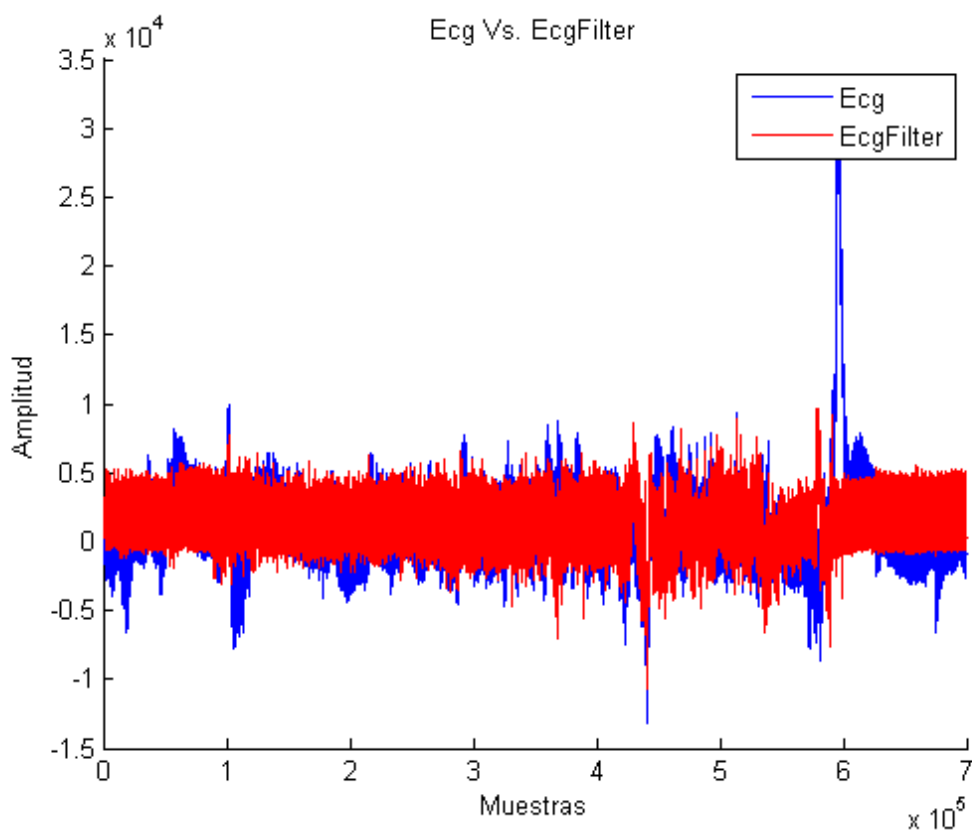
## Segundo método: Generación de la señal de movimiento de línea de base

Dentro de cada ciclo de la señal ECG existe un tramo de la onda donde el corazón se “apaga” eléctricamente y la misma se conforma solo de una señal mecánica, esto sucede entre las ondas P y Q.

En este segundo método nos basaremos de lo antes dicho para una vez obtenido la onda PQ poder obtener dichos puntos a lo largo de toda la señal de ECG1 provista por la cátedra. Luego utilizando la función *spline*, provista por Matlab, interpolaremos los puntos obtenidos y así generaremos la señal de línea de base. Finalmente realizaremos la resta de la señal de ECG con la señal generada y así buscaremos obtener la señal de ECG filtrada, mismo resultado que el obtenido en el primer método.

Para la realización de este segundo método contamos con la localización de cada latido de la señal ECG, provistos por la cátedra y el cual se adjunta a este informe dentro de la carpeta “señales.mat”. Por inspección de la señal de ECG observamos que en reiteradas ocasiones la distancia del segmento PQ (sobre el cual se colocó el punto a interpolar luego) al complejo QRS era de aproximadamente 80 muestras lo que equivale a 80nS motivo por el cual se utilizó este valor como offset. (*Ver código al final del ejercicio*)

A continuación se muestra el resultado obtenido, para el primer método, al aplicar el filtro a la señal de ECG1.



Se puede observar que el filtro utilizado cumplió de forma efectiva con el propósito para el cual fue realizado.

Tiempo de filtrado	0.8077 Segundos
--------------------	-----------------

A partir de aplicar el filtro propuesto se observó que el mismo tiene una demora en el filtrado de aproximadamente 0.8 Segundos.

A continuación se exponen en una tabla los tiempos de filtrado obtenidos con ambos métodos.

Método utilizado	Tiempo de procesamiento [Segundos]
Primer método	11.4735
Segundo método	0.8077

Del estudio realizado se puede concluir en que en tiempo de procesamiento el segundo método resulto ser más óptimo que el primero. Sin embargo, para el primer método se utilizó un filtro de mediana percentil 50 mientras que para el segundo método se debió realizar una señal a partir de la interpolación de los puntos obtenidos en la localización de la onda PQ. Para encontrar los puntos antes mencionados y poder realizar la interpolación, la cátedra, brindando la localización de los latidos de la señal de ECG, a partir de offset obtenido de forma visual se obtuvieron las abscisas de los puntos a interpolar, por lo que no se presentó ninguna inconveniencia en el filtrado por este método. Sin embargo, en el caso de que no hubiéramos contado con la localización de los latidos el primer método sería menos eficiente pero más rápido de implementar.

Además de las conclusiones antes mencionadas es importante destacar que el segundo método presenta un inconveniente ante el primer método debido a que éste tiene dependencia en cuanto a la localización del segmento PQ esto es debido a que el offset utilizado (en este caso de 80ns) se obtuvo de forma visual sobre el ECG, teniendo en cuenta que este valor no se mantiene del todo constante para toda la señal ECG en el tiempo, en cambio el filtro de mediana toma a todas las muestras y sobre el mismo no se estableció ningún parámetro de forma visual o iterativa.



## Ejercicio 4 – Códigos

A continuación se expone el código principal generado para la resolución del ejercicio

Ejercicio 4 - Código: Generación de la señal de movimiento de línea

```
QRS = QRS_loc1';
PQ = 80;

ejeX = zeros(length(QRS_loc1),1);
ejeY = zeros(length(QRS_loc1),1);

for k = 1 : length(QRS)

    if(QRS(k) > length(ECG1))
        ejeX = ejeX(1:k);
        ejeY = ejeY(1:k);
        break;
    end

    ejeX(k) = (QRS(k) - PQ);
    ejeY(k) = ECG1(QRS(k) - PQ);
end;

LineaBase = spline(ejeX,ejeY,1:length(ECG1));
```

## Bibliografía utilizada

- [ 1 ] Tratamiento de señales en tiempo discreto. Alan V. Oppenheim, Ronald W. Schafer. Ed. Pearson. Tercera edición.
- [ 3 ] Understanding Digital Signal Processing. Richard G. Lyons. Ed. Prentice Hall PTR
- [ 4 ] Tratamiento digital de señales. John G. Proakis y Dimitris G. Manolakis. 4ª Ed. Pearson Prentice Hall.