

# UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Santa Fe

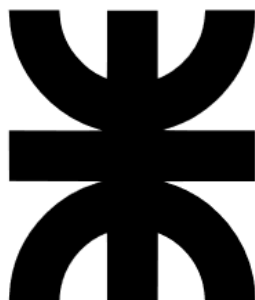
---

## Matemática Superior

*Ingeniería en Sistemas de Información*

### Trabajo Práctico I

### **Señales Biológicas: ECG**



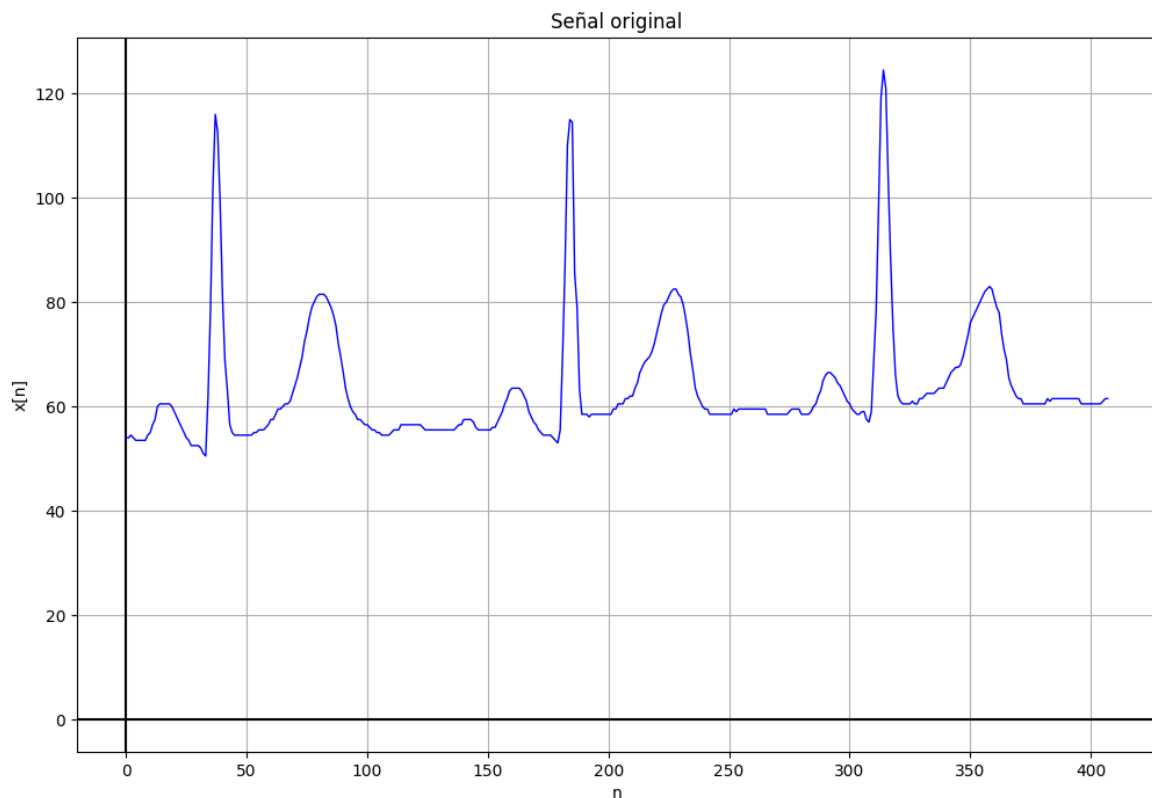
Alumno	Correo electrónico
Izaguirre, Ezequiel	ezequielmizaguirre@gmail.com
Pacheco Pilan, Federico Ignacio	fedepacheco2112@gmail.com
Rodríguez, Alejandro	rodriguezalejandro_@hotmail.com

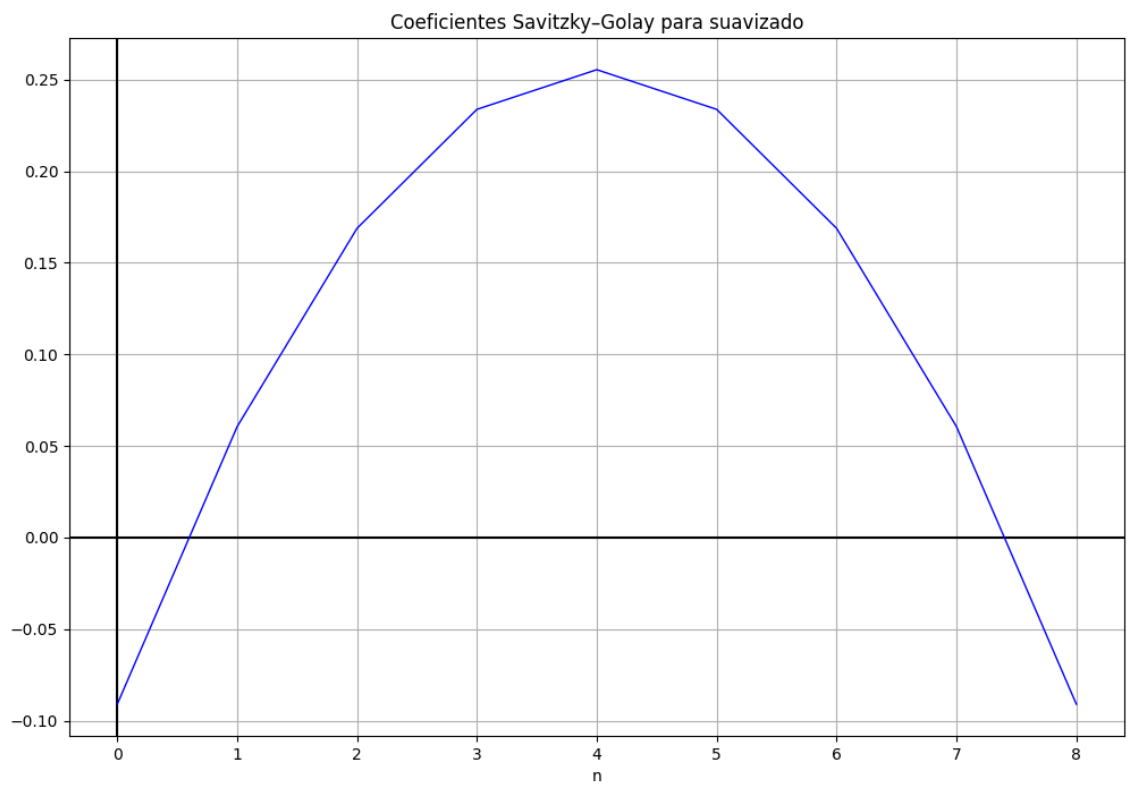
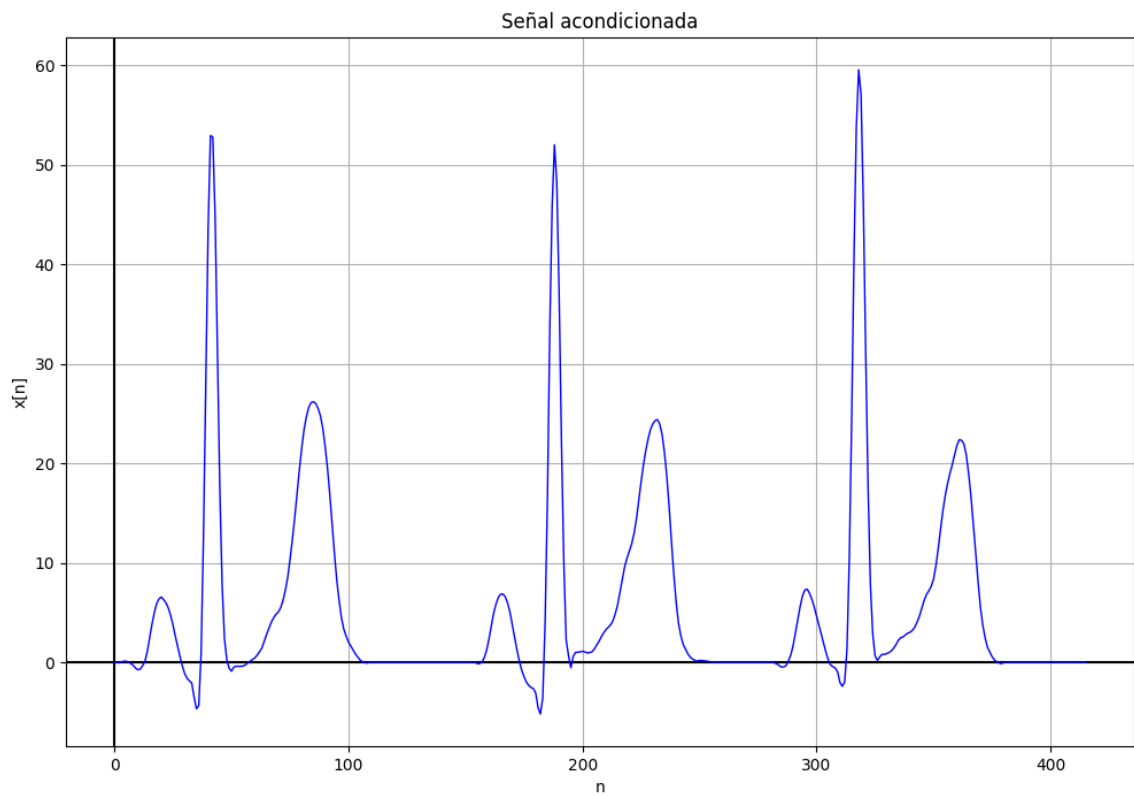
### Herramientas utilizadas

Para la realización del presente trabajo práctico se empleó el lenguaje de programación *Python* junto a las bibliotecas *numpy* (para realizar operaciones matemáticas) y *matplotlib* (para graficar funciones). El código fuente se adjunta al presente informe. Notar que para su ejecución, el archivo *cardio.csv* debe encontrarse en el mismo directorio.

## 1. Acondicionamiento de la señal

Tomando la señal original, se la colocó sobre el eje horizontal restándole la recta  $y[n] = 54 + 7 / 400 n$ , se eliminaron las ondas no correspondientes a la P, QRS o T mediante lomos de burro (colocados “a mano” por simplicidad) y por último se la suavizó convolucionándola (mediante *convolve()* de *numpy*) con una función similar a un “pinche”, obtenida de los coeficientes de Savitzky-Golar. El resultado se muestra a continuación.





## 2. Identificación de las ondas P, QRS y T

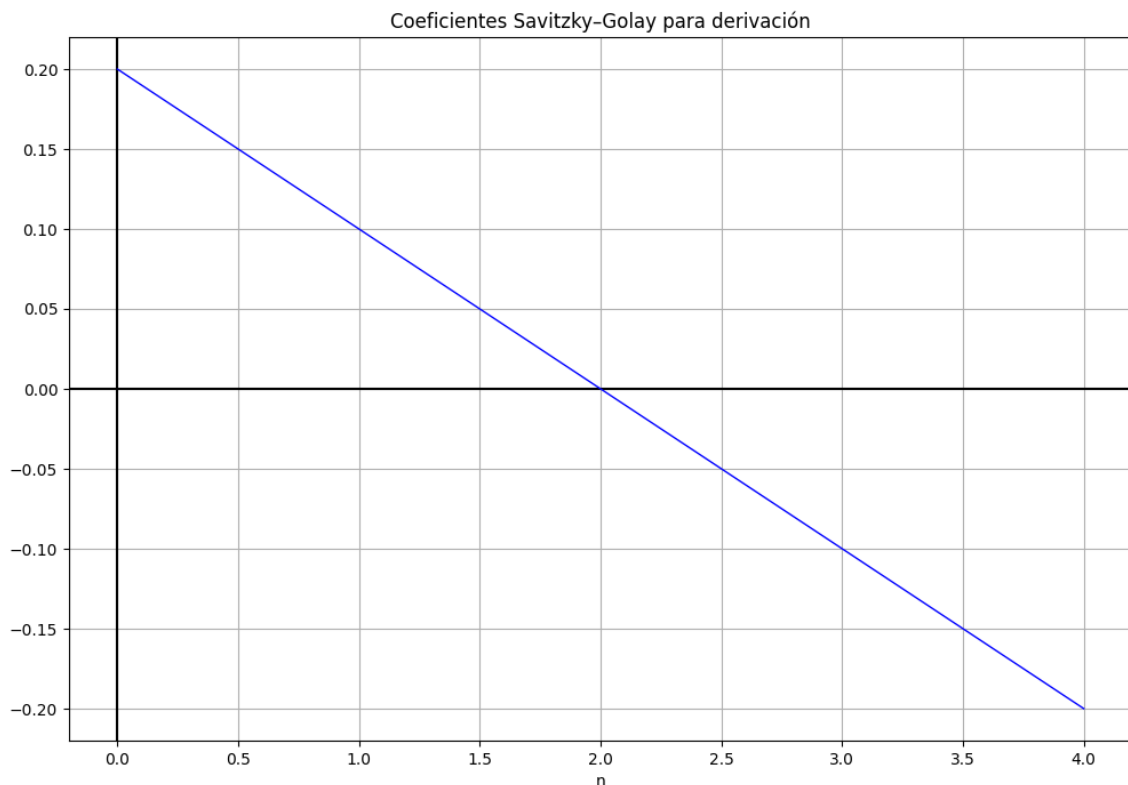
### 3.1. Aproximación de la derivada de la señal ECG:

En una primera instancia, la primera y segunda “derivada” de la señal fueron halladas convolucionando la misma una o dos veces con una función “doblete”, respectivamente:

$$\text{Doblete}[n] = \frac{\delta[n]}{2} - \frac{\delta[n-2]}{2}$$

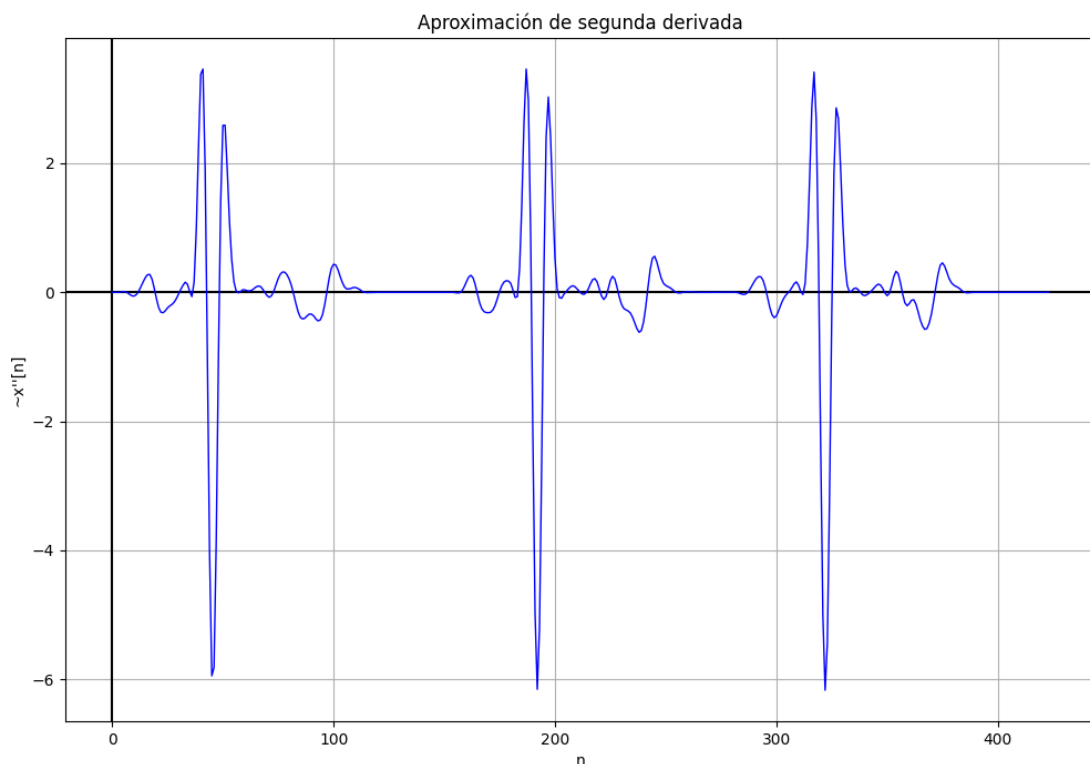
Luego, investigando una mejor manera de obtenerlas, se encontraron los coeficientes de Savitzky-Golay, los cuales mostraban un mejor suavizado en el resultado final de las derivadas respecto a la convolución con el doblete. A continuación, se muestra la función utilizada:

$$\text{Savitzky}[n] = \frac{2 \delta[n]}{10} + \frac{\delta[n-1]}{10} - \frac{\delta[n-3]}{10} - \frac{2 \delta[n-4]}{10}$$

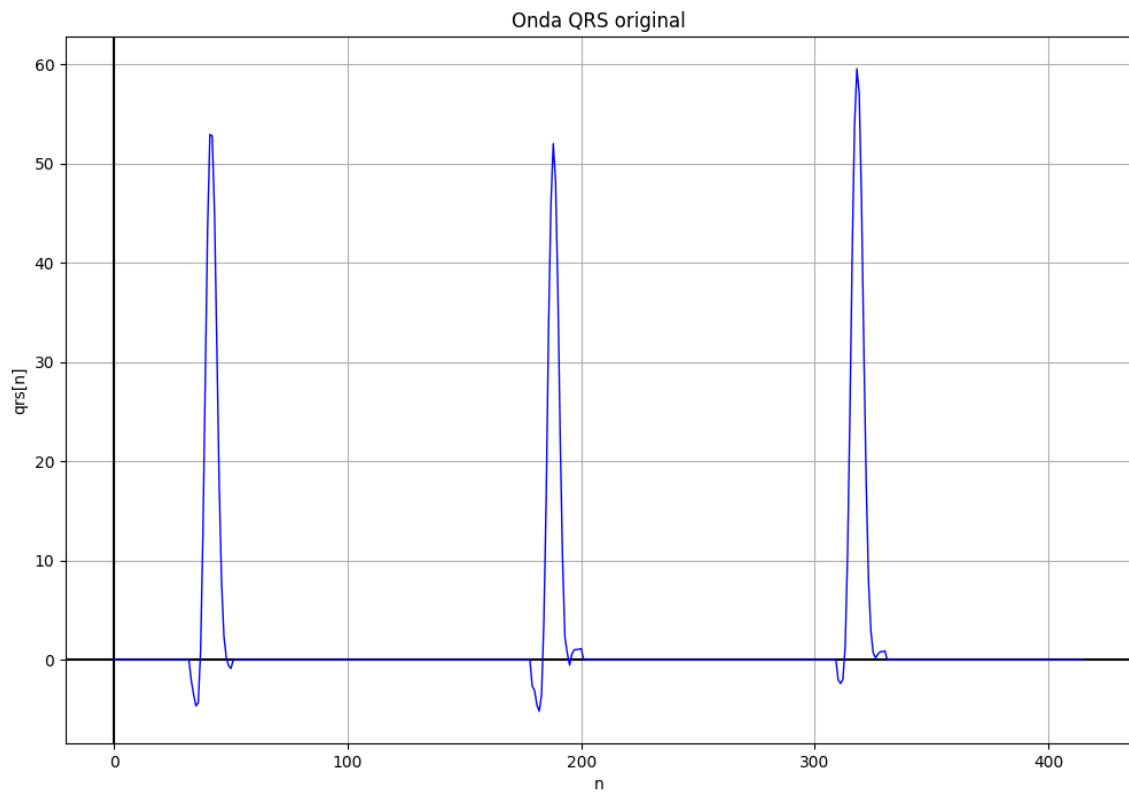
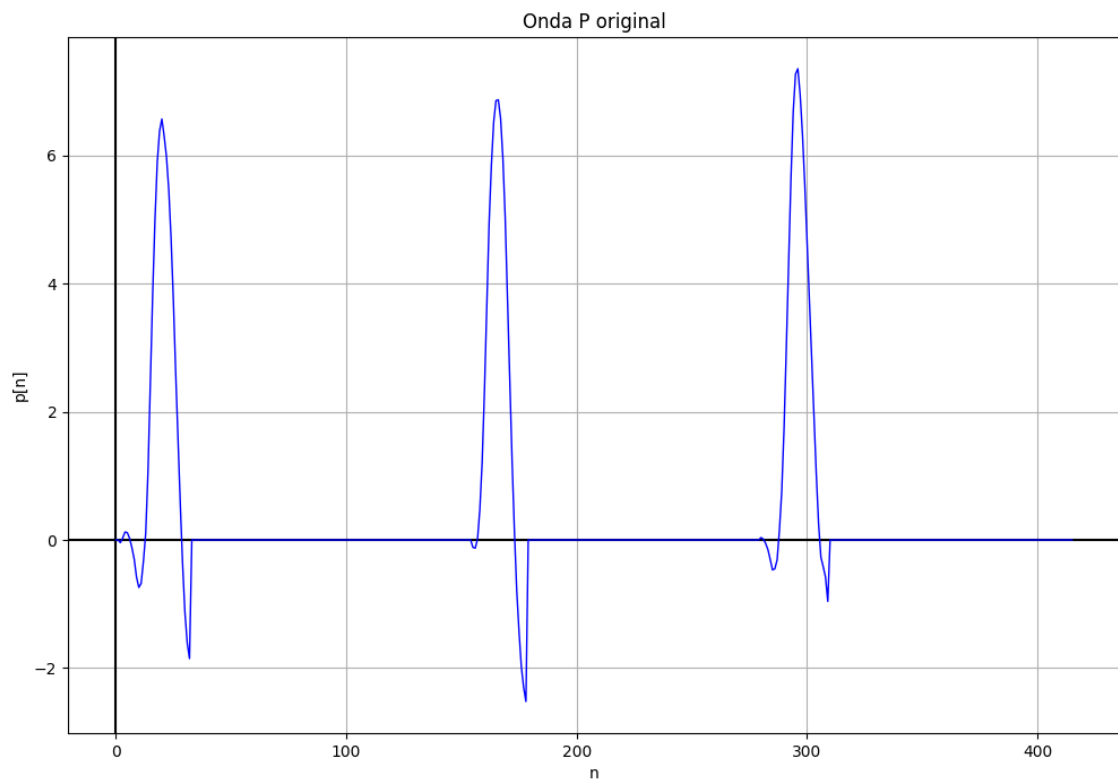


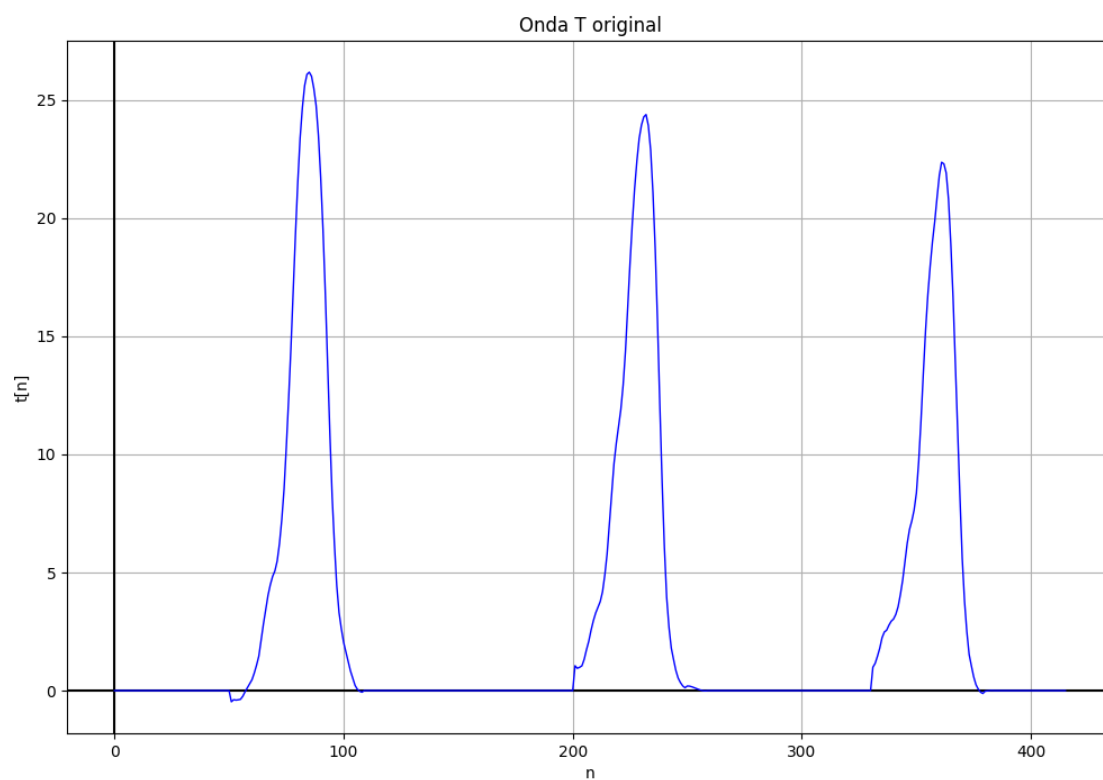
### 3.2. Aplicación de las derivadas:

Observando los gráficos que se anexan a continuación se tiene que, para identificar las tres ondas por medio de sus derivadas, la primera derivada es una mejor opción. Esto es así porque identificar una cresta y un valle similar a una onda sinusoidal en la primera derivada determina la etapa de crecimiento y decrecimiento de cada onda P, QRS o T.



De esta forma, las tres ondas separadas en el dominio del tiempo quedan de la siguiente manera (se lo hizo “a mano” por simplicidad):



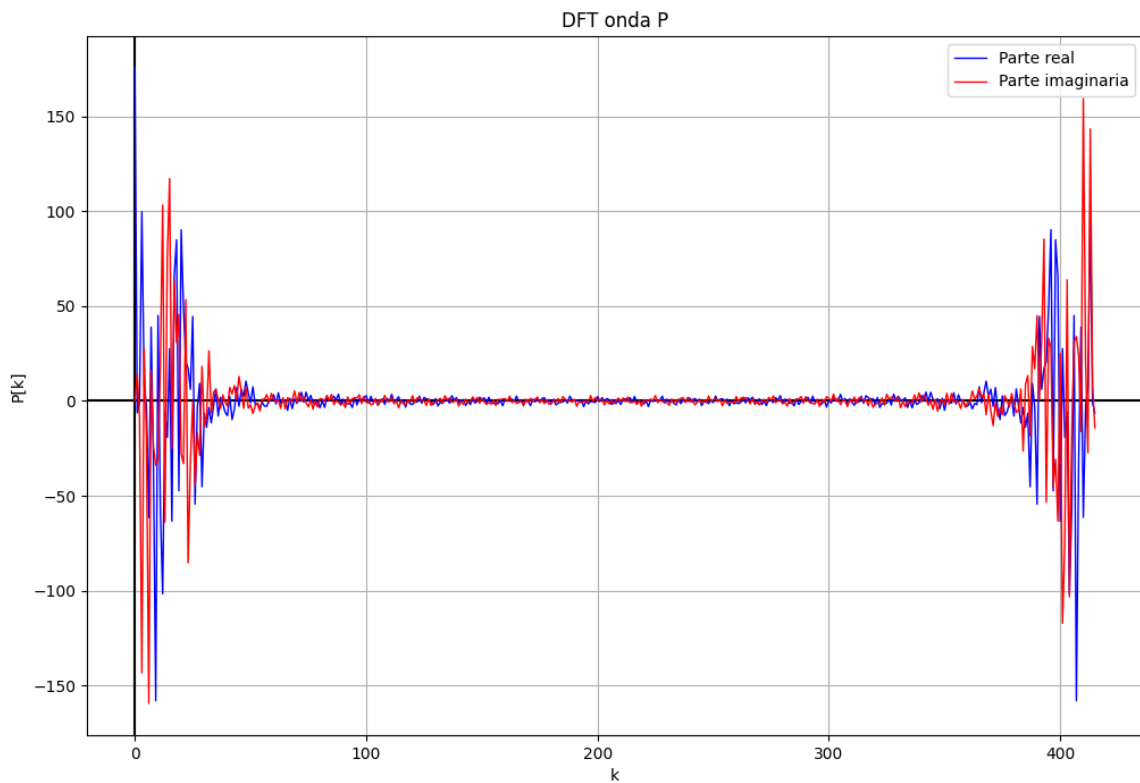


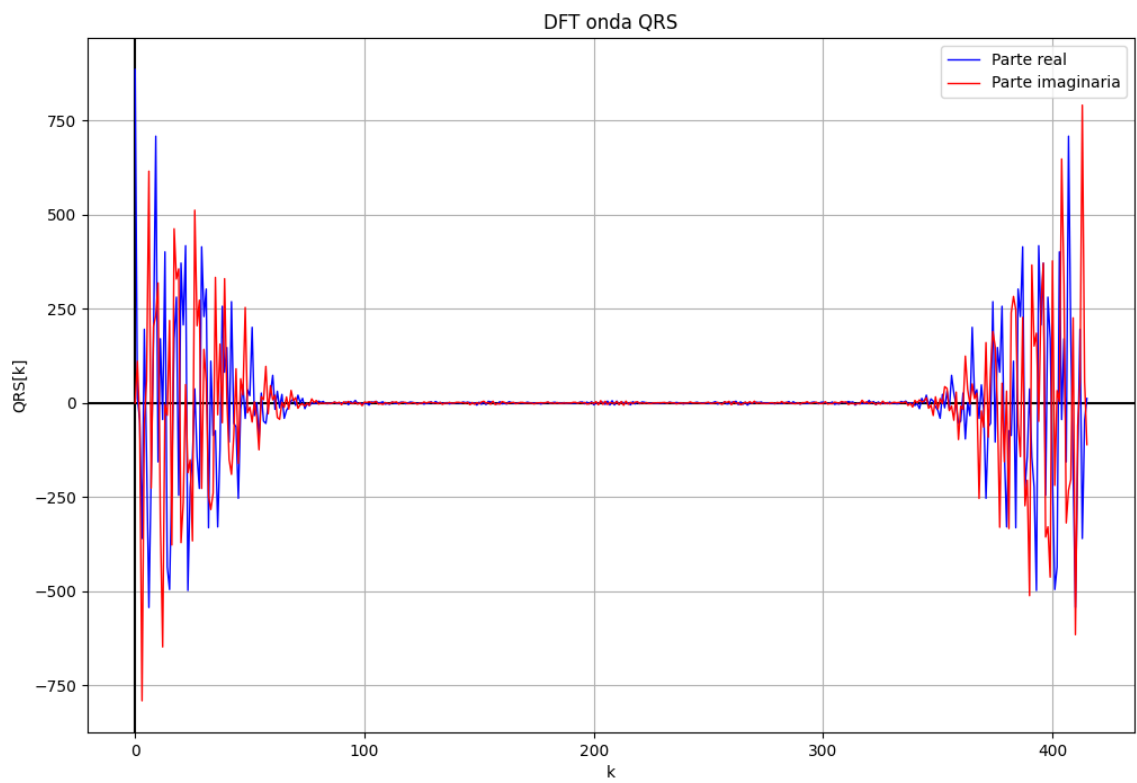
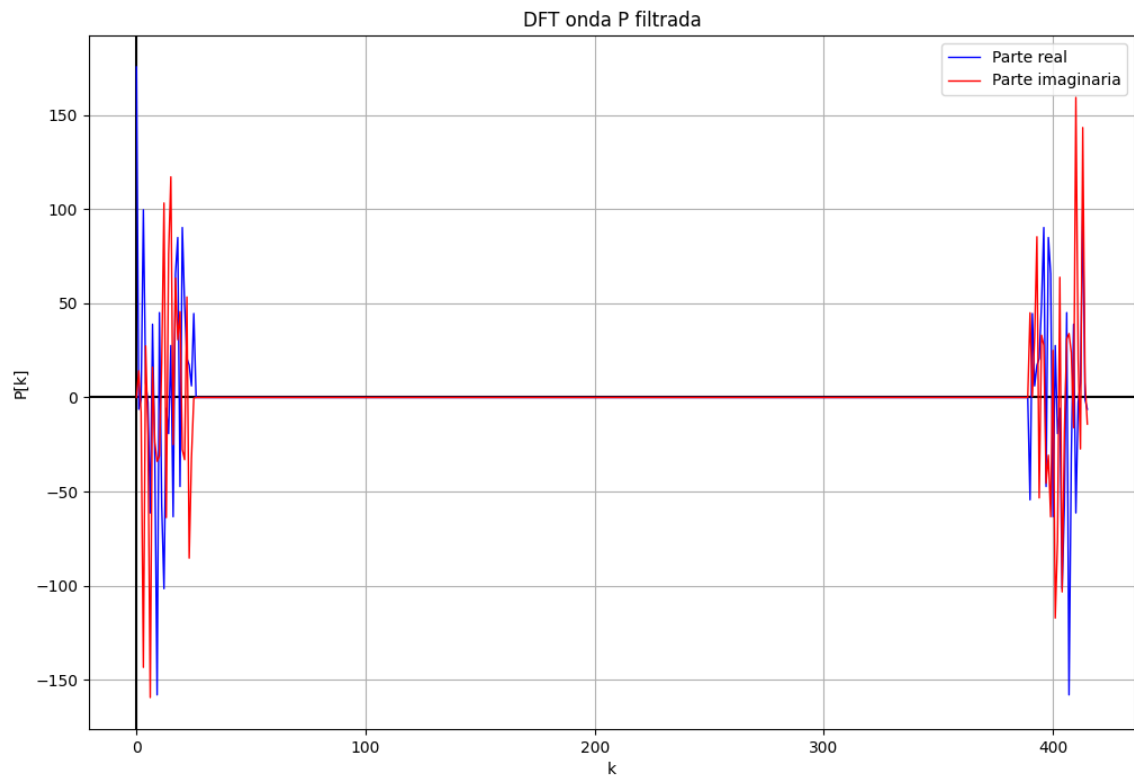


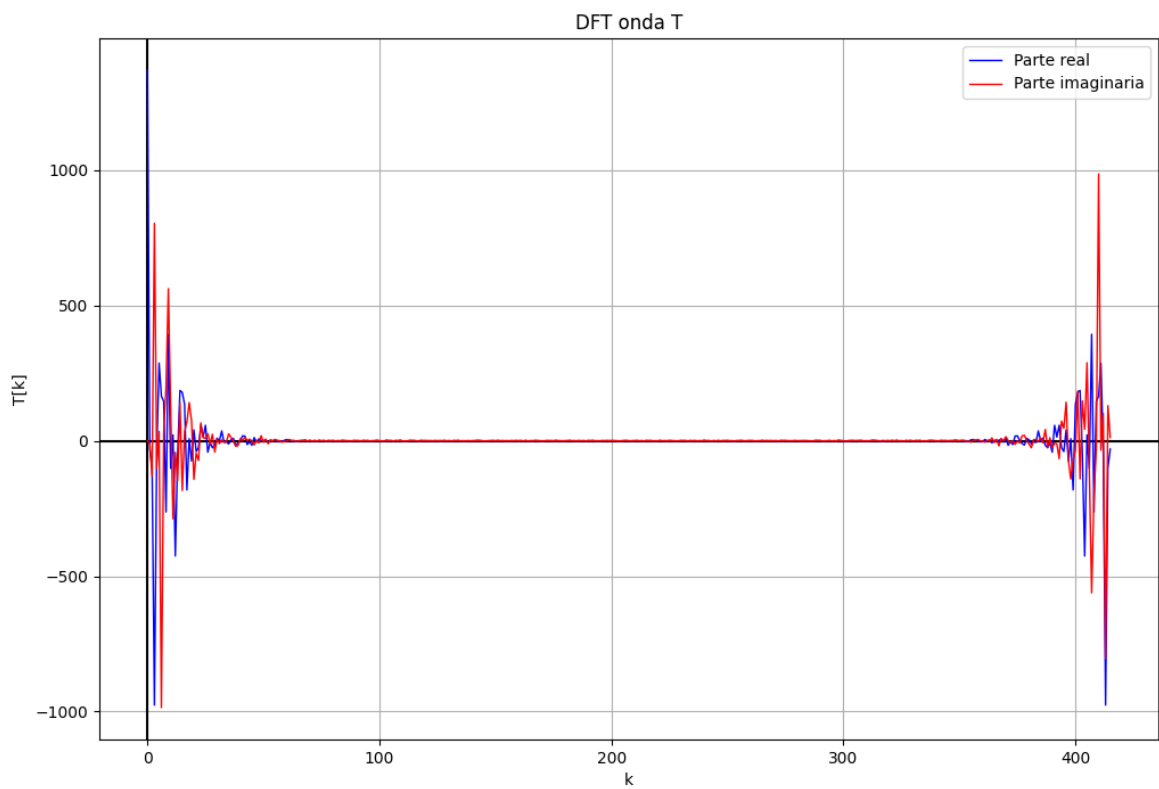
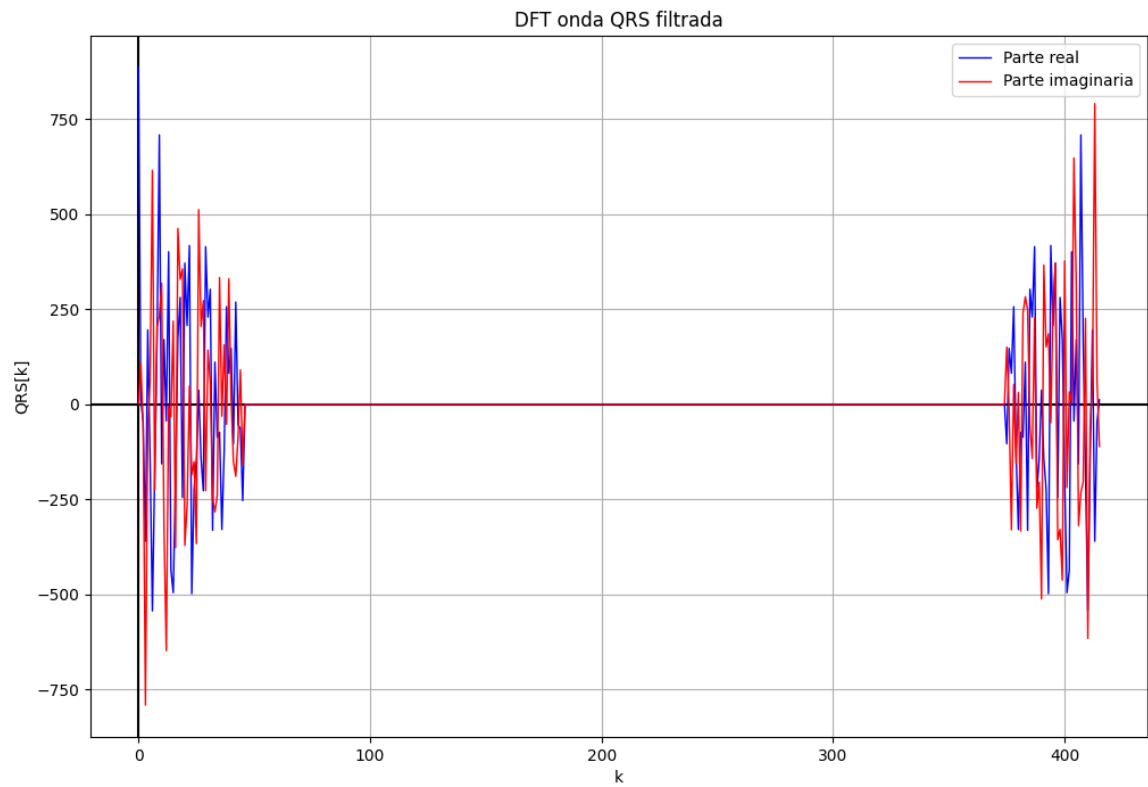
## 4. Reconstrucción de la señal de ECG por separación de ondas

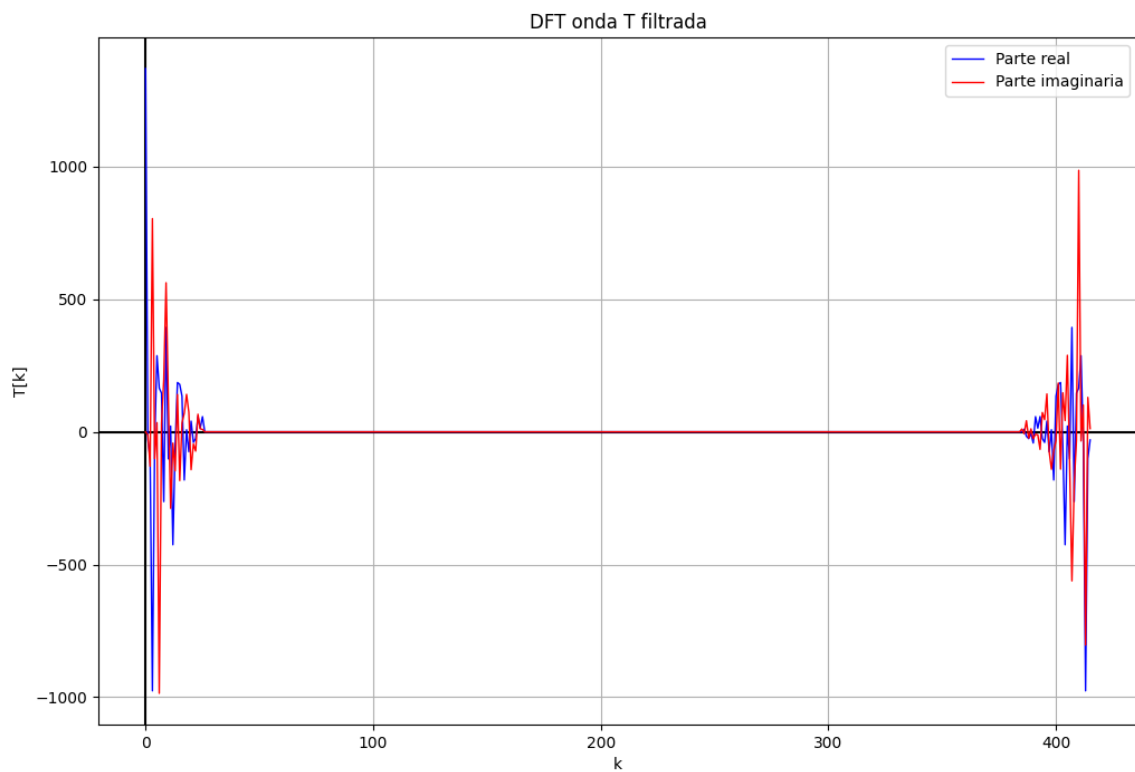
### 4.1. Recorte en frecuencia:

Se aplicó la transformada discreta de fourier utilizando `numpy.fft.fft()` para posteriormente con lomos de burro simular un filtro pasabanda “inverso” ideal (unión entre un filtro pasa-bajo “a la izquierda” y un pasa-alto “a la derecha”). El rango de frecuencia a recortar en cada caso fue elegido “a mano” basándose en la forma de los gráficos de la transformada original. Así, se tiene:



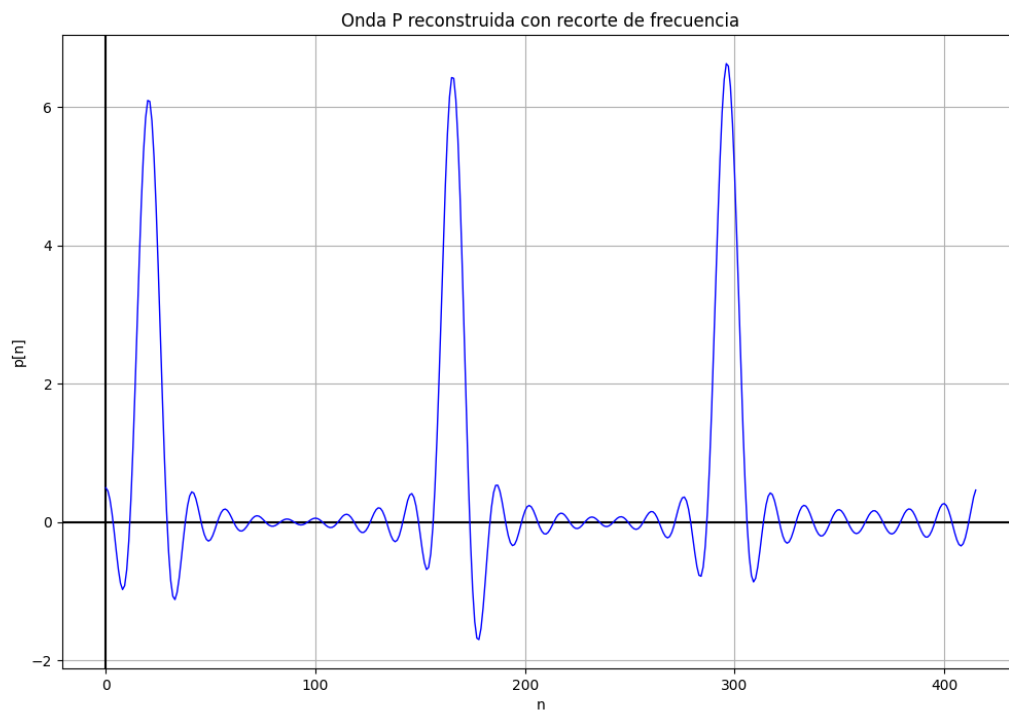


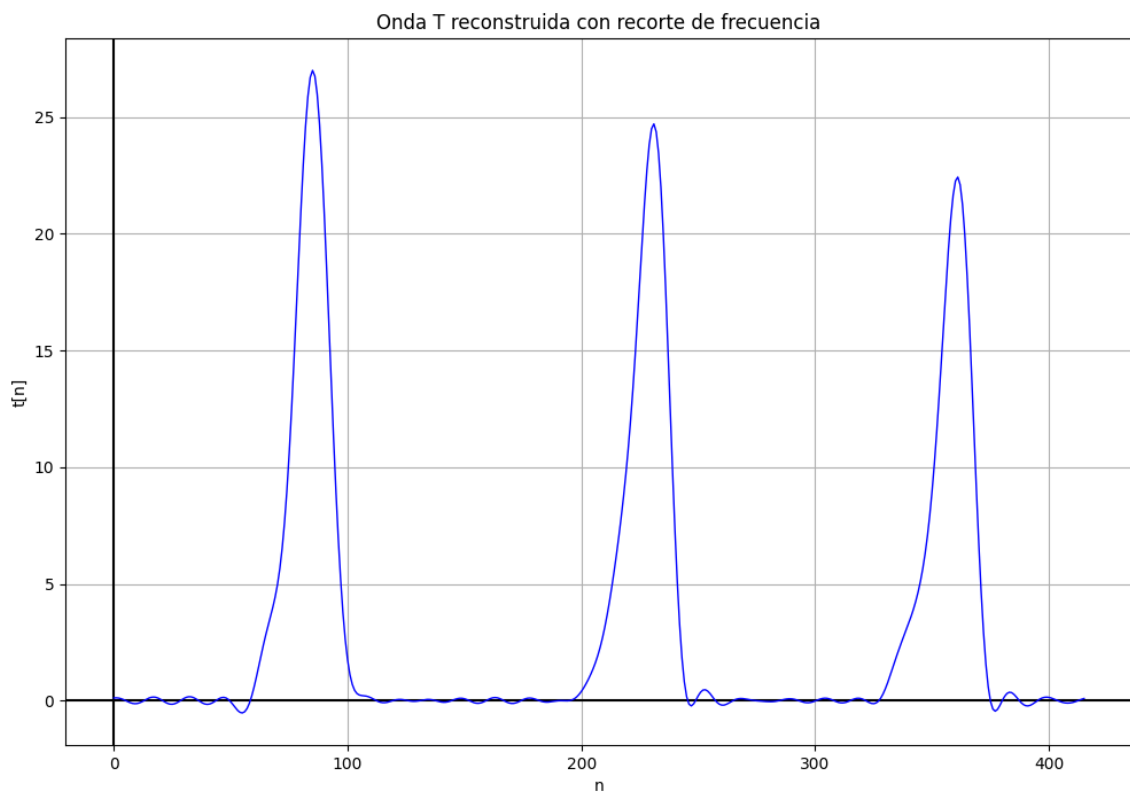
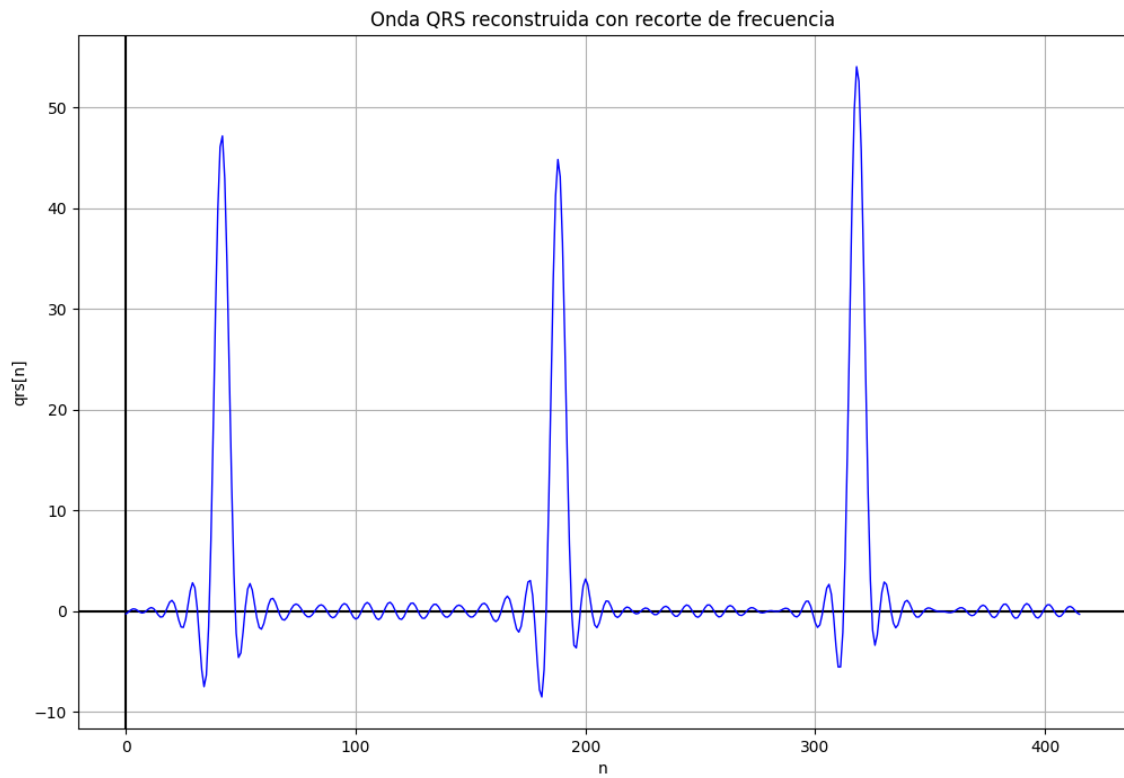




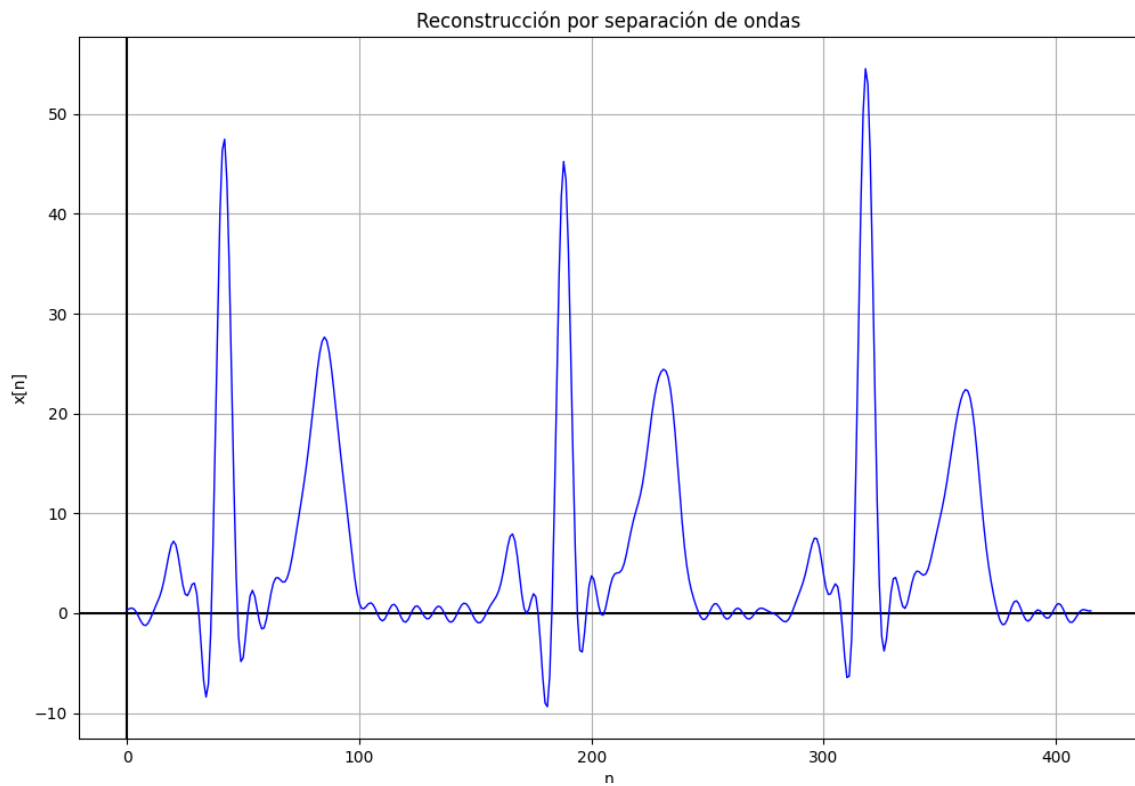
#### **4.2. Reconstrucción:**

Empleando `numpy.fft.iff()` para cada onda individual se obtuvo:





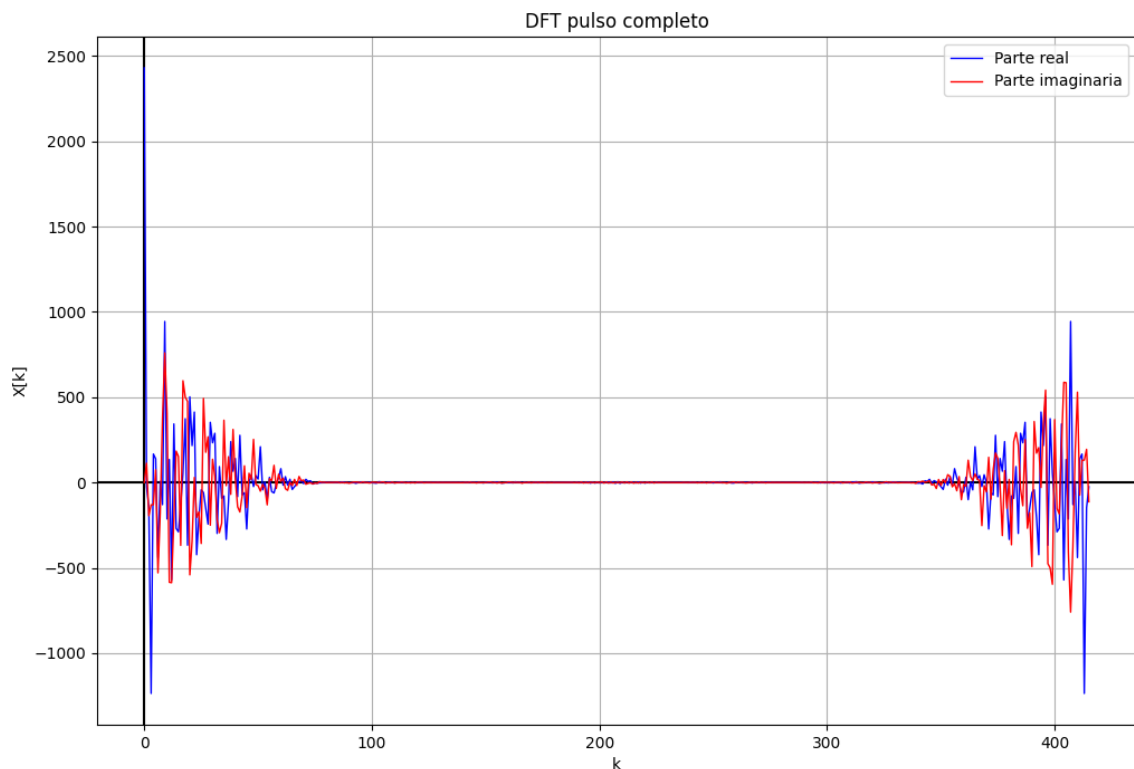
Como suma de las tres ondas reconstruidas en el tiempo se obtuvo la siguiente aproximación de la señal original:

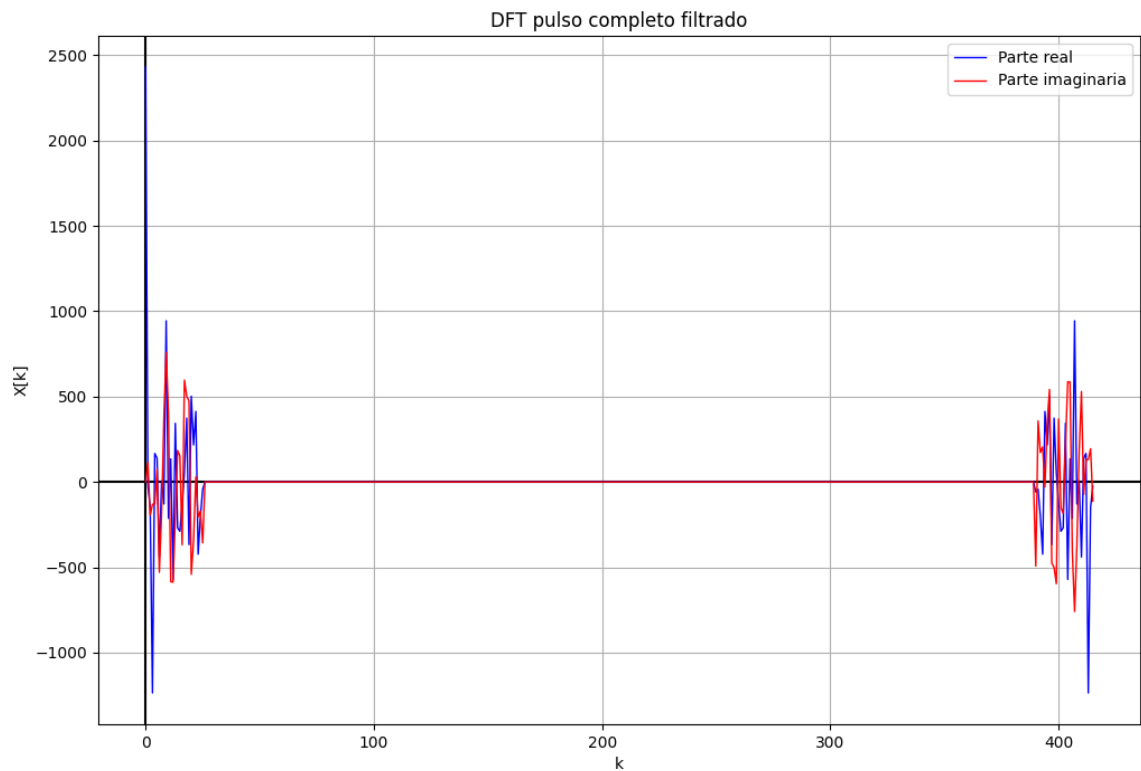


## 5. Reconstrucción de la señal de ECG sin separación de ondas

### 5.1. Recorte en frecuencia:

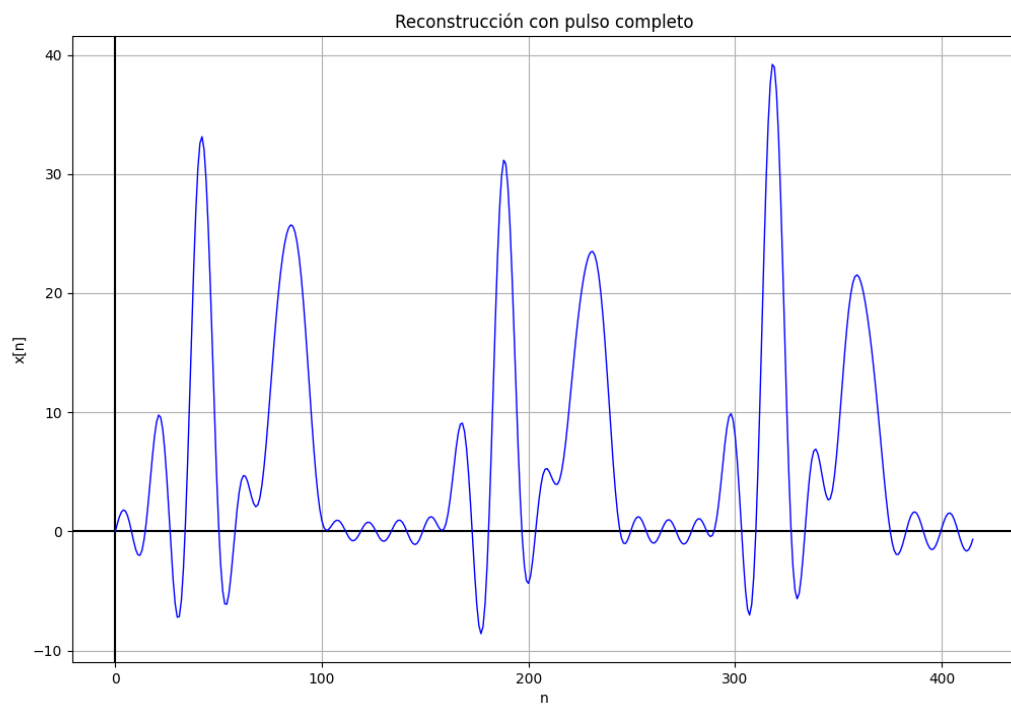
Para este caso, luego de aplicada la transformada discreta de fourier, se escogió el filtro pasabanda “inverso” de tal modo que su valor “piso” en el rango de frecuencias es el mínimo de entre los valores “piso” utilizados en los filtros de cada onda y su valor “techo” es el máximo de entre los valores “techo” utilizados en los filtros de cada onda. Así, se tiene:





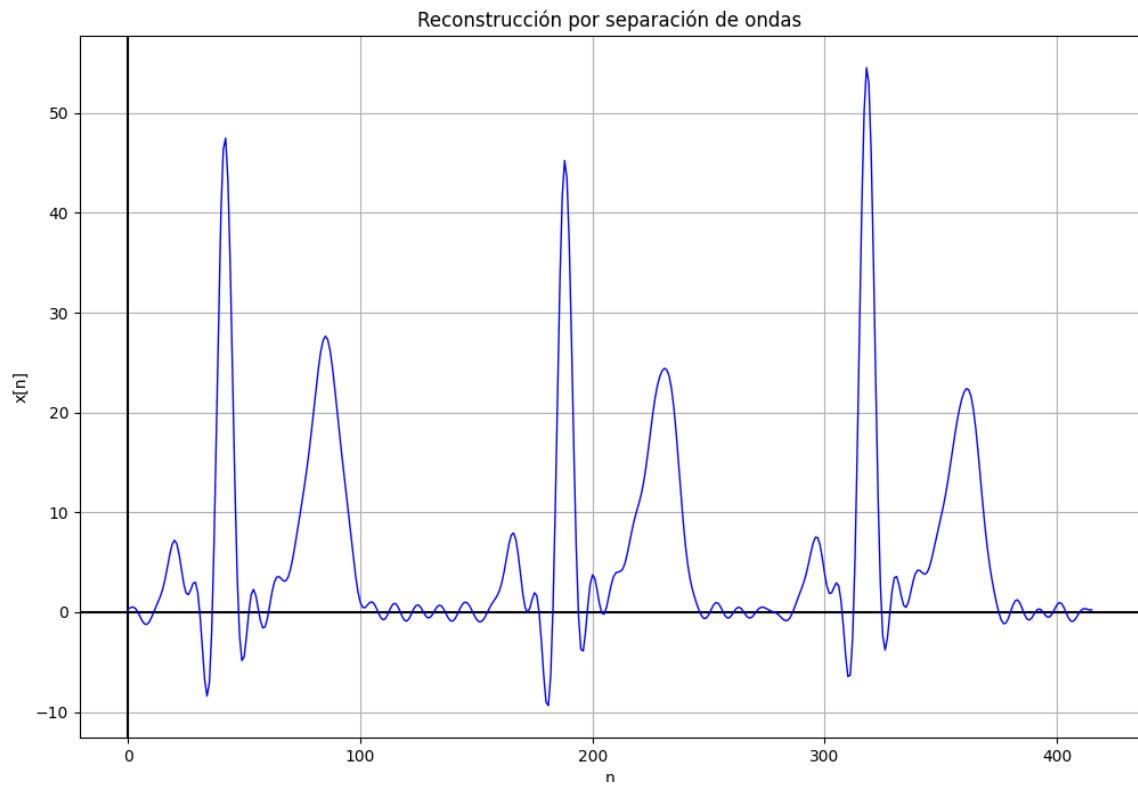
## 5.2. Reconstrucción:

Volviendo al dominio temporal, la reconstrucción queda:





Se coloca nuevamente la reconstrucción por separación de ondas para facilitar la comparación entre ambos casos:



## 6. Conclusiones del filtrado y reconstrucción

Tanto en el caso del filtrado de las ondas separadas como en el de filtrado del pulso completo, se realizó un recorte en frecuencia fundamentalmente en las zonas donde la transformada se mantenía cerca del eje de abscisas, puesto que si no en la reconstrucción la señal se distorsionaba “demasiado”.

Se observó que filtrar en frecuencia cada onda por separado, reconstruir y sumar es, en general, una mejor alternativa si se quiere preservar la calidad de la señal en mayor medida. Esto tiene la desventaja de que el costo de realizar tales operaciones es mayor computacionalmente hablando. Por otra parte, filtrar el pulso completo es más simple y barato, pero tiene la desventaja de degenerar más la señal reconstruida. Esto puede observarse, por ejemplo, en cómo las ondas QRS pierden amplitud, de tal suerte que quedan de tamaño comparable a las ondas T en los dos primeros ciclos.

## 7. Webgrafía

- <https://www.geeksforgeeks.org/working-csv-files-python/>
- <https://stackoverflow.com/questions/12201928/python-open-gives-filenotfounderror-io-error-errno-2-no-such-file-or-directory>
- <https://stackoverflow.com/questions/5137497/find-current-directory-and-files-directory>
- <https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/>
- <https://stackoverflow.com/questions/17109608/change-figure-size-and-figure-format-in-matplotlib>
- <https://stackoverflow.com/questions/4056768/how-to-declare-array-of-zeros-in-python-or-an-array-of-a-certain-size>
- <https://numpy.org/doc/stable/reference/generated/numpy.convolve.html>
- <https://numpy.org/doc/stable/reference/generated/numpy.fft.fft.html>
- <https://numpy.org/doc/stable/reference/generated/numpy.fft.ifft.html>
- [https://www.researchgate.net/figure/Electrocardiograma-con-sus-ondas-intervalos-segmentos-y-su-relacion-con-el-ciclo\\_fig2\\_333649418](https://www.researchgate.net/figure/Electrocardiograma-con-sus-ondas-intervalos-segmentos-y-su-relacion-con-el-ciclo_fig2_333649418)
- <https://docs.python.org/3/library/operator.html>
- <https://stackoverflow.com/questions/22276066/how-to-plot-multiple-functions-on-the-same-figure-in-matplotlib>
- [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.legend.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html)
- [https://chem.libretexts.org/Bookshelves/Analytical\\_Chemistry/Supplemental\\_Modules\\_\(Analytical\\_Chemistry\)/Analytical\\_Sciences\\_Digital\\_Library/JASDL/Courseware/Introduction\\_to\\_Signals\\_and\\_Noise/04\\_Signal-to-Noise\\_Enhancement/03\\_Digital\\_Filtering/05\\_Convolution-Based\\_Smoothing](https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Analytical_Sciences_Digital_Library/JASDL/Courseware/Introduction_to_Signals_and_Noise/04_Signal-to-Noise_Enhancement/03_Digital_Filtering/05_Convolution-Based_Smoothing)
- [https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golay\\_filter#Tables\\_of\\_selected\\_convolution\\_coefficients](https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golay_filter#Tables_of_selected_convolution_coefficients)