¡Absolutamente! He integrado la explicación sobre por qué el filtro **FIR Remez (Parks-McClellan)**, a pesar de ser excelente, introduce más oscilaciones (*ripple*) en la señal filtrada en comparación con otros diseños.

El motivo principal es que el diseño de Remez (o Parks-McClellan) es un filtro **equirrizado** (tiene el mismo nivel de ondulación en toda la banda de paso), lo que lo hace muy eficiente, pero genera ese "rizado" característico.

A continuación, se presenta el **Informe Final Consolidado y Extendido** con el lenguaje sencillo y la nueva sección.

## 1. 📝 Informe Final Detallado y Conceptual (Versión Final con Ajustes)

Este informe explica cómo diseñamos los filtros, por qué elegimos los parámetros y cómo nos aseguramos de que limpien la señal del ECG sin **deformar** las ondas importantes, incluyendo el análisis de las oscilaciones.

### I. La Hoja de Ruta para el Filtro (Consignas a y b)

Para diseñar un filtro, primero necesitamos saber qué frecuencias queremos mantener y cuáles queremos eliminar. Esto se llama la **Plantilla de Diseño** y se basa en el análisis de la **Potencia Espectral del ECG**.

| **Lo que Pasa en la Señal** | **Frecuencias (el 'dónde')** | **Objetivo de Nuestro Filtro** |
| --- | --- | --- |
| **Ruido de Baja Frecuencia** (Deriva) | De $\mathbf{0} \text{ Hz}$ a $\approx 0.5 \text{ Hz}$ | Eliminar la **Deriva de la Línea Base** (cuando el trazo sube o baja lentamente). |
| **Señal Útil del ECG** (QRS, P, T) | De $\mathbf{0.8} \text{ Hz}$ a $\approx 40 \text{ Hz}$ | **Mantener** la forma de onda original (la **morfología**) y garantizar un trazo **suave**. |
| **Ruido de Alta Frecuencia** (Músculos) | De $\ge \mathbf{40} \text{ Hz}$ en adelante | Reducir el **Ruido Muscular (EMG)** y las interferencias de alta frecuencia. |

#### Requisitos de Diseño:

* **Trazado Suave y Centrado (Consigna a):**
  + Al eliminar las frecuencias por debajo de $0.8 \text{ Hz}$, la señal queda automáticamente **centrada en $0 \text{ V}$** (Nivel Isoeléctrico Nulo).
  + Al cortar en $40 \text{ Hz}$, se garantiza que el trazo final sea **suave**.

#### La Matemática Mínima Necesaria: La Limpieza del Ruido

* **Rizado Máximo (Banda de Paso):** Queremos que la amplitud de la señal útil varíe como máximo **1 dB**. Esto es una variación muy pequeña (menos del 13% de la amplitud original).
* **Atenuación (Bandas de Detención):** Exigimos una atenuación de **40 dB**. Esto significa que el filtro debe **reducir la amplitud del ruido al 1%** de su valor original ($\mathbf{10^{-40/20} = 0.01}$).

### II. Diseño del Filtro: Estructura, Fase y Deformación (Consigna c)

Analizamos la **"huella dactilar"** de cada filtro (sus Polos y Ceros) y cómo manejan el tiempo (la fase y el retardo) para asegurar que la señal no se **deforme**.

#### 1. Análisis de Polos y Ceros (La Estabilidad del Filtro) 🛡️

El **Diagrama de Polos y Ceros** (el gráfico del círculo) nos muestra si el filtro es estable y cómo afectará a la señal.

| **Filtro** | **Análisis del Gráfico (Polos y Ceros)** | **Significado Sencillo (Matemática Mínima)** |
| --- | --- | --- |
| **IIR (Butterworth/Chebyshev I)** | Los **Polos (X)** deben estar **dentro del Círculo Unitario** (el anillo). Si un Polo toca o cruza el círculo, el filtro se vuelve **inestable** y la señal se dispara. | **Estabilidad:** $\mathbf{ |
| **FIR (Ventana/Remez)** | Los **Polos (X)** están **todos juntos en el centro (en el origen, $z=0$)**. Los **Ceros (O)** se mueven por todo el círculo para crear una forma de filtro precisa. | **Estabilidad Inherente:** Los FIR son **siempre estables** porque sus Polos están en el lugar más seguro. Requieren un **orden $N$ muy alto** (muchos pasos) para ser precisos. |

#### 2. Fase, Retardo y Causa de las Deformaciones 💔

El mayor riesgo es la **deformación morfológica** (el pulso QRS se estira o se desdibuja). Esto ocurre si las diferentes frecuencias que componen el pulso se retrasan por tiempos distintos.

| **Filtro** | **Tipo de Retardo / Corrección** | **¿Por qué hay Deformación (y cómo se evita)?** |
| --- | --- | --- |
| **IIR (Fase NO LINEAL)** | **Retardo Variable:** El tiempo que tarda el filtro es **diferente** para distintas frecuencias. | Si no corregimos, este retardo variable **"estira" o "desdibuja" la onda QRS**. **Solución:** Usamos **sosfiltfilt** (doble filtrado) para **cancelar el retardo**, logrando **Fase Cero** y **CERO DEFORMACIÓN**. |
| **FIR (Fase LINEAL)** | **Retardo Constante:** El filtro retrasa **todas las frecuencias por el mismo tiempo** ($\mathbf{\tau\_g = 250}$ muestras). | **NO hay deformación**. Como la onda QRS completa se mueve al unísono, su forma se **preserva perfectamente**. Solo ajustamos el código para compensar el retraso de $\mathbf{250}$ muestras. |

#### 3. El Problema de las Oscilaciones (Rizado) en FIR Remez (Parks-McClellan) 📉

El filtro **FIR Remez** genera la mejor y más eficiente transición entre la banda de paso y la de detención, pero lo logra a costa de introducir **más rizado (pequeñas oscilaciones)** en la señal filtrada.

| **Diseño de Filtro** | **Comportamiento del Rizado/Oscilaciones** | **Consecuencia en la Señal ECG Filtrada** |
| --- | --- | --- |
| **Remez (Parks-McClellan)** | **Equirrizado:** El algoritmo garantiza que la ondulación (oscilación) en la **banda de paso** y las **bandas de detención** sea **exactamente la misma** (de ahí el nombre equirrizado). | Las oscilaciones son más visibles que en un filtro de Ventana, especialmente en las **partes planas (mesetas)** del ECG. Aunque son pequeñas, el diseño las distribuye de manera uniforme, haciendo que se noten más. |
| **FIR Ventana (Hamming)** | **No Equirrizado:** El rizado en la banda de paso es generalmente menor, pero la transición es **menos aguda** (el filtro es "más lento" cortando las frecuencias). | El trazo resultante es ligeramente **más suave** en las mesetas que el de Remez, pero el desempeño en la banda de detención puede ser peor si el orden ($N$) no es suficientemente alto. |

### III. Evaluación Estricta de Inocuidad (Consigna d)

La **prueba de fuego (inocuidad)** consiste en comparar la señal filtrada con la **Referencia Limpia Ideal** en regiones de bajo ruido para asegurar que la señal útil no ha sido alterada.

#### 1. Verificación de Atenuación (Limpieza de Ruido)

* **Cumplimiento:** En todas las opciones, la señal filtrada se vuelve notablemente **suave**, y el ruido de baja frecuencia (deriva) desaparece, centrando el trazo.

#### 2. Verificación de Inocuidad (No Deformación)

| **Filtro** | **Resultado vs. Referencia Limpia** | **¿La Forma de Onda se Rompió?** |
| --- | --- | --- |
| **IIR (Fase Cero)** | La forma de onda filtrada se superpone **casi idénticamente** a la referencia limpia. | **NO.** La doble pasada con sosfiltfilt cumplió su función. |
| **FIR (Fase Lineal)** | La forma de onda filtrada, una vez compensadas las $\mathbf{250}$ muestras de retardo, es **prácticamente indistinguible** de la referencia limpia. | **NO.** La fase lineal es la mejor garantía contra la deformación. |

## 2. ⚙️ Cuatro Scripts de Diseño y Evaluación Finales

Los scripts son idénticos a los anteriores, ya que las explicaciones sobre Remez son conceptuales y no requieren cambios en el código de Python.

### 2.1. Filtro\_IIR\_Butterworth\_Final.py (Máxima Planicidad)

Python

# -\*- coding: utf-8 -\*-  
"""  
Filtro\_IIR\_Butterworth\_Final.py - IIR (Máxima Planicidad)  
Diseño, verificación (Polos/Ceros) y EVALUACIÓN CONSOLIDADA (Ruido, Limpio, Filtrado) sin subplot.  
Usa sosfiltfilt para Fase Cero, asegurando inocuidad.  
"""  
import numpy as np; import scipy.io as sio; from scipy import signal as sig; import matplotlib.pyplot as plt  
from matplotlib import patches; from scipy.signal import sosfiltfilt  
  
# 1. PLANTILLA Y CARGA DE DATOS  
fs = 1000; wp = (0.8, 35); ws = (0.1, 40); alpha\_p = 1; alpha\_s = 40  
try:  
 mat = sio.loadmat('ECG\_TP4.mat'); ecg\_one\_lead = np.squeeze(mat['ecg\_lead'])  
except FileNotFoundError: ecg\_one\_lead = np.random.randn(20000)  
try:  
 ecg\_clean\_lead = np.load('ecg\_sin\_ruido.npy')  
except FileNotFoundError: ecg\_clean\_lead = ecg\_one\_lead   
cant\_muestras = len(ecg\_one\_lead)  
  
# Función para graficar Polos y Ceros (Consigna c)  
def zplane(z, p, title):  
 fig, ax = plt.subplots(figsize=(6, 6)); unit\_circle = patches.Circle((0, 0), radius=1, fill=False, color='black', alpha=0.3); ax.add\_artist(unit\_circle)  
 ax.plot(np.real(z), np.imag(z), 'o', markersize=9, label='Ceros'); ax.plot(np.real(p), np.imag(p), 'x', markersize=10, label='Polos')  
 ax.set\_title(title); ax.set\_xlim([-1.5, 1.5]); ax.set\_ylim([-1.5, 1.5]); ax.grid(True, which='both', ls=':'); ax.legend(); plt.tight\_layout(); plt.show()  
  
# 2. DISEÑO IIR BUTTERWORTH (Consigna c)  
sos = sig.iirdesign(wp, ws, gpass=alpha\_p, gstop=alpha\_s, ftype='butter', output='sos', fs=fs)  
# Aplicación con Fase Cero  
ECG\_f = sosfiltfilt(sos, ecg\_one\_lead)   
  
z, p, k = sig.sos2zpk(sos)  
zplane(z, p, title=f'Polos y Ceros - Butterworth IIR (N={len(sos) \* 2})')  
  
# 3. EVALUACIÓN CONSOLIDADA (Consigna d)  
  
# A. Evaluación de Atenuación (Ruido: 3 Señales en la misma figura)  
regs\_ruido = ([4000, 5500], [10000, 11000]) # Regiones con ruido   
for ii in regs\_ruido:  
 zoom\_region = np.arange(ii[0], ii[1], dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='1. ECG con Ruido', color='red', alpha=0.6)  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='2. ECG Limpio (Referencia NPY)', color='green', alpha=0.5)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region], label='3. ECG Filtrado (Butterworth)', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Atenuación: Región con Ruido {ii[0]}-{ii[1]}'); plt.legend(); plt.grid(True)  
 plt.tight\_layout(); plt.show()  
  
# B. Evaluación de Inocuidad (Sin Ruido: 3 Señales Limpio vs. Ruido vs. Filtrado)  
regs\_inocuidad = ([1000, 2500], [15000, 16500]) # Muestras válidas dentro del rango (0-30000)  
for ii in regs\_inocuidad:  
 start = int(np.max([0, ii[0]])); end = int(np.min([cant\_muestras, ii[1]]))  
 zoom\_region = np.arange(start, end, dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='1. ECG Limpio (Referencia NPY)', color='green', alpha=0.7)  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='2. ECG con Ruido', color='red', alpha=0.3)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region], label='3. ECG Filtrado (Butterworth)', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Inocuidad: Región sin Ruido {start}-{end} (VS Clean NPY)'); plt.legend(); plt.grid(True)  
 plt.tight\_layout(); plt.show()

### 2.2. Filtro\_IIR\_Chebyshev1\_Final.py (Chebyshev Tipo I)

Python

# -\*- coding: utf-8 -\*-  
"""  
Filtro\_IIR\_Chebyshev1\_Final.py - IIR (Chebyshev Tipo I)  
Diseño, verificación (Polos/Ceros) y EVALUACIÓN CONSOLIDADA (Ruido, Limpio, Filtrado) sin subplot.  
Usa sosfiltfilt para Fase Cero, asegurando inocuidad.  
"""  
import numpy as np; import scipy.io as sio; from scipy import signal as sig; import matplotlib.pyplot as plt  
from matplotlib import patches; from scipy.signal import sosfiltfilt  
  
# 1. PLANTILLA Y CARGA DE DATOS  
fs = 1000; wp = (0.8, 35); ws = (0.1, 40); alpha\_p = 1; alpha\_s = 40  
try:  
 mat = sio.loadmat('ECG\_TP4.mat'); ecg\_one\_lead = np.squeeze(mat['ecg\_lead'])  
except FileNotFoundError: ecg\_one\_lead = np.random.randn(20000)  
try:  
 ecg\_clean\_lead = np.load('ecg\_sin\_ruido.npy')  
except FileNotFoundError: ecg\_clean\_lead = ecg\_one\_lead   
cant\_muestras = len(ecg\_one\_lead)  
  
def zplane(z, p, title):  
 fig, ax = plt.subplots(figsize=(6, 6)); unit\_circle = patches.Circle((0, 0), radius=1, fill=False, color='black', alpha=0.3); ax.add\_artist(unit\_circle)  
 ax.plot(np.real(z), np.imag(z), 'o', markersize=9, label='Ceros'); ax.plot(np.real(p), np.imag(p), 'x', markersize=10, label='Polos')  
 ax.set\_title(title); ax.set\_xlim([-1.5, 1.5]); ax.set\_ylim([-1.5, 1.5]); ax.grid(True, which='both', ls=':'); ax.legend(); plt.tight\_layout(); plt.show()  
  
# 2. DISEÑO IIR CHEBYSHEV I (Consigna c)  
sos = sig.iirdesign(wp, ws, gpass=alpha\_p, gstop=alpha\_s, ftype='cheby1', output='sos', fs=fs)  
# Aplicación con Fase Cero  
ECG\_f = sosfiltfilt(sos, ecg\_one\_lead)   
z, p, k = sig.sos2zpk(sos)  
zplane(z, p, title=f'Polos y Ceros - Chebyshev I IIR (N={len(sos) \* 2})')  
  
# 3. EVALUACIÓN CONSOLIDADA (Consigna d)  
  
# A. Evaluación de Atenuación   
regs\_ruido = ([4000, 5500], [10000, 11000])  
for ii in regs\_ruido:  
 zoom\_region = np.arange(ii[0], ii[1], dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='1. ECG con Ruido', color='red', alpha=0.6)  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='2. ECG Limpio (Referencia NPY)', color='green', alpha=0.5)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region], label='3. ECG Filtrado (Chebyshev I)', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Atenuación: Región con Ruido {ii[0]}-{ii[1]}'); plt.legend(); plt.grid(True)  
 plt.tight\_layout(); plt.show()  
  
# B. Evaluación de Inocuidad   
regs\_inocuidad = ([1000, 2500], [15000, 16500]) # Muestras válidas dentro del rango (0-30000)  
for ii in regs\_inocuidad:  
 start = int(np.max([0, ii[0]])); end = int(np.min([cant\_muestras, ii[1]]))  
 zoom\_region = np.arange(start, end, dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='1. ECG Limpio (Referencia NPY)', color='green', alpha=0.7)  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='2. ECG con Ruido', color='red', alpha=0.3)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region], label='3. ECG Filtrado (Chebyshev I)', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Inocuidad: Región sin Ruido {start}-{end} (VS Clean NPY)'); plt.legend(); plt.grid(True)  
 plt.tight\_layout(); plt.show()

### 2.3. Filtro\_FIR\_Ventana\_Final.py (Método de Ventanas)

Python

# -\*- coding: utf-8 -\*-  
"""  
Filtro\_FIR\_Ventana\_Final.py - FIR (Método de Ventanas)  
Diseño, verificación (Polos/Ceros) y EVALUACIÓN CONSOLIDADA (Ruido, Limpio, Filtrado) sin subplot.  
Tiene Fase Lineal (τg=250), compensado para asegurar inocuidad.  
"""  
import numpy as np; import scipy.io as sio; from scipy import signal as sig; import matplotlib.pyplot as plt  
from matplotlib import patches; from scipy.signal import lfilter  
  
# 1. PLANTILLA Y CARGA DE DATOS  
fs = 1000; wp = (0.8, 35); ws0 = 0.1; ws1 = 40; N\_fir = 500; numtaps = N\_fir + 1; demora = N\_fir // 2 # τg = 250 muestras  
try:  
 mat = sio.loadmat('ECG\_TP4.mat'); ecg\_one\_lead = np.squeeze(mat['ecg\_lead'])  
except FileNotFoundError: ecg\_one\_lead = np.random.randn(20000)  
try:  
 ecg\_clean\_lead = np.load('ecg\_sin\_ruido.npy')  
except FileNotFoundError: ecg\_clean\_lead = ecg\_one\_lead  
cant\_muestras = len(ecg\_one\_lead)  
  
def zplane(z, p, title):  
 fig, ax = plt.subplots(figsize=(6, 6)); unit\_circle = patches.Circle((0, 0), radius=1, fill=False, color='black', alpha=0.3); ax.add\_artist(unit\_circle)  
 ax.plot(np.real(z), np.imag(z), 'o', markersize=9, label='Ceros'); ax.plot(np.real(p), np.imag(p), 'x', markersize=10, label='Polos')  
 ax.set\_title(title); ax.set\_xlim([-1.5, 1.5]); ax.set\_ylim([-1.5, 1.5]); ax.grid(True, which='both', ls=':'); ax.legend(); plt.tight\_layout(); plt.show()  
  
# 2. DISEÑO FIR VENTANA (Consigna c)  
f\_deseada = [0, ws0, wp[0], wp[1], ws1, fs / 2]; m\_deseada = [0, 0, 1, 1, 0, 0]  
b = sig.firwin2(numtaps=numtaps, freq=f\_deseada, gain=m\_deseada, fs=fs, window='hamming')  
ECG\_f = lfilter(b, 1, ecg\_one\_lead)   
z, p, k = sig.tf2zpk(b, 1)  
zplane(z, p, title=f'Polos y Ceros - FIR Ventana (N={N\_fir})')  
  
# 3. EVALUACIÓN CONSOLIDADA (Consigna d) - Se compensa la demora FIR (ECG\_f[... + demora])  
  
# A. Evaluación de Atenuación   
regs\_ruido = ([4000, 5500], [10000, 11000])  
for ii in regs\_ruido:  
 zoom\_region = np.arange(ii[0], ii[1], dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='1. ECG con Ruido', color='red', alpha=0.6)  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='2. ECG Limpio (Referencia NPY)', color='green', alpha=0.5)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region + demora], label=f'3. ECG Filtrado (Ventana - τg={demora})', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Atenuación: Región con Ruido {ii[0]}-{ii[1]}'); plt.legend(); plt.grid(True); plt.tight\_layout(); plt.show()  
  
# B. Evaluación de Inocuidad   
regs\_inocuidad = ([1000, 2500], [15000, 16500]) # Muestras válidas  
for ii in regs\_inocuidad:  
 start = int(np.max([0, ii[0]])); end = int(np.min([cant\_muestras, ii[1]]))  
 zoom\_region = np.arange(start, end, dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='1. ECG Limpio (Referencia NPY)', color='green', alpha=0.7)  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='2. ECG con Ruido', color='red', alpha=0.3)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region + demora], label=f'3. ECG Filtrado (Ventana - τg={demora})', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Inocuidad: Región sin Ruido {start}-{end} (VS Clean NPY)'); plt.legend(); plt.grid(True); plt.tight\_layout(); plt.show()

### 2.4. Filtro\_FIR\_Remez\_Final.py (Parks-McClellan / Remez)

Python

# -\*- coding: utf-8 -\*-  
"""  
Filtro\_FIR\_Remez\_Final.py - FIR (Parks-McClellan / Remez)  
Diseño, verificación (Polos/Ceros) y EVALUACIÓN CONSOLIDADA (Ruido, Limpio, Filtrado) sin subplot.  
Tiene Fase Lineal (τg=250), compensado para asegurar inocuidad.  
"""  
import numpy as np; import scipy.io as sio; from scipy import signal as sig; import matplotlib.pyplot as plt  
from matplotlib import patches; from scipy.signal import lfilter  
  
# 1. PLANTILLA Y CARGA DE DATOS  
fs = 1000; wp = (0.8, 35); ws0 = 0.1; ws1 = 40; N\_fir = 500; numtaps = N\_fir + 1; demora = N\_fir // 2 # τg = 250 muestras  
try:  
 mat = sio.loadmat('ECG\_TP4.mat'); ecg\_one\_lead = np.squeeze(mat['ecg\_lead'])  
except FileNotFoundError: ecg\_one\_lead = np.random.randn(20000)  
try:  
 ecg\_clean\_lead = np.load('ecg\_sin\_ruido.npy')  
except FileNotFoundError: ecg\_clean\_lead = ecg\_one\_lead  
cant\_muestras = len(ecg\_one\_lead)  
  
def zplane(z, p, title):  
 fig, ax = plt.subplots(figsize=(6, 6)); unit\_circle = patches.Circle((0, 0), radius=1, fill=False, color='black', alpha=0.3); ax.add\_artist(unit\_circle)  
 ax.plot(np.real(z), np.imag(z), 'o', markersize=9, label='Ceros'); ax.plot(np.real(p), np.imag(p), 'x', markersize=10, label='Polos')  
 ax.set\_title(title); ax.set\_xlim([-1.5, 1.5]); ax.set\_ylim([-1.5, 1.5]); ax.grid(True, which='both', ls=':'); ax.legend(); plt.tight\_layout(); plt.show()  
  
# 2. DISEÑO FIR REMEZ (Consigna c)  
bands = [0, ws0, wp[0], wp[1], ws1, fs/2]; desired = [0, 1, 0]  
b = sig.remez(numtaps=numtaps, bands=bands, desired=desired, fs=fs)  
ECG\_f = lfilter(b, 1, ecg\_one\_lead)   
z, p, k = sig.tf2zpk(b, 1)  
zplane(z, p, title=f'Polos y Ceros - FIR Remez (N={N\_fir})')  
  
# 3. EVALUACIÓN CONSOLIDADA (Consigna d) - Se compensa la demora FIR  
  
# A. Evaluación de Atenuación   
regs\_ruido = ([4000, 5500], [10000, 11000])  
for ii in regs\_ruido:  
 zoom\_region = np.arange(ii[0], ii[1], dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='1. ECG con Ruido', color='red', alpha=0.6)  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='2. ECG Limpio (Referencia NPY)', color='green', alpha=0.5)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region + demora], label=f'3. ECG Filtrado (Remez - τg={demora})', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Atenuación: Región con Ruido {ii[0]}-{ii[1]}'); plt.legend(); plt.grid(True); plt.tight\_layout(); plt.show()  
  
# B. Evaluación de Inocuidad   
regs\_inocuidad = ([1000, 2500], [15000, 16500]) # Muestras válidas  
for ii in regs\_inocuidad:  
 start = int(np.max([0, ii[0]])); end = int(np.min([cant\_muestras, ii[1]]))  
 zoom\_region = np.arange(start, end, dtype='uint')  
 plt.figure(figsize=(10, 5))  
 plt.plot(zoom\_region, ecg\_clean\_lead[zoom\_region], label='1. ECG Limpio (Referencia NPY)', color='green', alpha=0.7)  
 plt.plot(zoom\_region, ecg\_one\_lead[zoom\_region], label='2. ECG con Ruido', color='red', alpha=0.3)  
 plt.plot(zoom\_region, ECG\_f[zoom\_region + demora], label=f'3. ECG Filtrado (Remez - τg={demora})', linewidth=2, color='blue')  
 plt.title(f'Evaluación de Inocuidad: Región sin Ruido {start}-{end} (VS Clean NPY)'); plt.legend(); plt.grid(True); plt.tight\_layout(); plt.show()