

U.T.N. Facultad Regional Paraná

Ingeniería Electrónica



Técnicas Digitales III

2026

Examen Parcial Integrador PDS

Alumnos: Cafaro Alejandro, Guiglioni Lautaro, Penon Marcos

Mails: Alejandrocafaro@alu.frp.utn.edu.ar , LautaroGuiglioni@alu.frp.utn.edu.ar ,
Marcospenon@alu.frp.utn.edu.ar

Docentes: Ing. Yarce Gustavo, Ing. Dachary Alejandro

Fecha de Presentación: 3 de Julio de 2026.

Paraná, Entre Ríos.

II_ Resumen Ejecutivo

El presente trabajo aborda el diseño e implementación de un sistema de filtrado digital aplicado al procesamiento de señales provenientes de un Monitor Continuo de Glucosa (CGM) simulado. La señal de glucosa en sangre es una variable fisiológica de dinámica lenta, con componentes de interés por debajo de 0.5 Hz, afectada por deriva DC del sensor, ruido de cuantización y artefactos de alta frecuencia. El objetivo principal es diseñar una cadena de procesamiento digital que elimine estas perturbaciones preservando la información clínicamente relevante de la señal.

Se diseñaron e implementaron dos filtros digitales complementarios. El primero es un filtro pasa-bajos de arquitectura FIR con fase lineal estricta, destinado al suavizado de ruido de alta frecuencia con frecuencia de corte en 0.5 Hz. El segundo es un filtro pasa-altos de arquitectura IIR basado en el prototipo analógico Butterworth de orden 2, diseñado mediante la Transformada Bilineal, orientado a la eliminación de la deriva DC del sensor con frecuencia de corte en 0.01 Hz. Ambos filtros operan a una frecuencia de muestreo de 10 Hz, implementados sobre la plataforma ESP32.

El diseño fue verificado mediante simulación en MATLAB, confirmando el cumplimiento de las especificaciones de atenuación y la estabilidad del sistema. Adicionalmente, se desarrolló un agente de decisión basado en reglas que recomienda automáticamente el tipo de filtro más adecuado en función de restricciones de diseño como memoria disponible, requerimiento de fase lineal, nivel de ruido y capacidad de cómputo. El sistema fue validado con múltiples escenarios de prueba, demostrando consistencia entre las recomendaciones del agente y el análisis teórico realizado.

III _ Índice

II_ Resumen Ejecutivo	2
IV_ Marco Teórico	4
4.1 Filtros Digitales	4
4.2 Filtros FIR (Finite Impulse Response)	4
4.3 Filtros IIR (Infinite Impulse Response)	4
4.4 Métodos de diseño	5
4.4.1 Diseño de filtros IIR — Transformada Bilineal	5
4.4.2 Prototipos analógicos	5
4.4.3 Diseño de filtros FIR — Método de ventanas	5
4.5 Fase lineal	6
4.6 Estabilidad de filtros digitales	6
4.7 Estructuras de implementación	6
4.8 Sistema de decisión — Agente basado en reglas	7
V_ Análisis del Problema y Definición de Especificaciones	7
5.1 Identificación y caracterización de la señal	7
5.2 Especificaciones técnicas del sistema de filtrado	8
Tabla de Requerimientos Técnicos	9
VI_ Diseño de filtros	10
Filtro FIR Pasa bajos	10
Filtro IIR Pasa altos	14
VII_ Simulación de filtros MATLAB	18
FIR Pasa Bajo Simulado	18
IIR Pasa Alto Simulado	22
Comparaciones Filtros	26
VIII_ Agentes de IA	27
IX_ Implementación Embebida	27
X_ Conclusiones	27
XI_ Bibliografía	27

IV_ Marco Teórico

4.1 Filtros Digitales

Un filtro digital es un sistema discreto en el tiempo que procesa una secuencia de entrada $x[n]$ para producir una secuencia de salida $y[n]$, modificando selectivamente el contenido espectral de la señal. A diferencia de los filtros analógicos, operan sobre muestras numéricas y pueden implementarse en procesadores digitales de señales, microcontroladores o computadoras. Los filtros digitales se clasifican en dos grandes familias según la naturaleza de su respuesta al impulso: filtros de Respuesta Finita al Impulso (FIR) y filtros de Respuesta Infinita al Impulso (IIR).

4.2 Filtros FIR (Finite Impulse Response)

Los filtros de Respuesta Finita al Impulso, conocidos como filtros FIR, producen una salida que depende exclusivamente de muestras presentes y pasadas de la señal de entrada. No existe realimentación de la salida hacia la entrada, lo que significa que la respuesta al impulso tiene una duración finita y bien determinada. Cada muestra de salida se calcula como una combinación lineal ponderada de un número finito de muestras de entrada, donde los pesos de esa combinación son los coeficientes del filtro.

La principal ventaja de los filtros FIR es su capacidad de garantizar fase lineal estricta. Esto significa que todas las componentes frecuenciales de la señal son retardadas exactamente el mismo número de muestras al atravesar el filtro, preservando intacta la forma de onda temporal de la señal. Esta propiedad es crítica en aplicaciones médicas y fisiológicas, donde la distorsión de la morfología de la señal podría inducir errores en el diagnóstico o el análisis.

Otra ventaja fundamental es su estabilidad incondicional. Al no poseer realimentación, un filtro FIR es siempre estable independientemente de sus coeficientes, eliminando la necesidad de verificación de estabilidad como paso de diseño.

Su principal desventaja es el elevado orden que generalmente requieren para lograr transiciones espectrales selectivas. Un orden elevado implica mayor cantidad de operaciones por muestra, mayor consumo de memoria y mayor latencia, lo que puede ser una limitación en sistemas embebidos con recursos restringidos.

4.3 Filtros IIR (Infinite Impulse Response)

Los filtros de Respuesta Infinita al Impulso, conocidos como filtros IIR, producen una salida que depende tanto de muestras pasadas de la entrada como de muestras pasadas de la propia salida. Esta realimentación genera una respuesta al impulso de duración teóricamente infinita, de ahí su nombre.

La principal ventaja de los filtros IIR es su eficiencia computacional. Para una misma especificación de selectividad frecuencial, un filtro IIR requiere un orden significativamente menor que un filtro FIR equivalente, lo que se traduce en menos operaciones por muestra, menor uso de memoria y menor latencia. Esto los hace especialmente atractivos para implementaciones en tiempo real con recursos de cómputo limitados.

Sin embargo, los filtros IIR presentan dos desventajas importantes. En primer lugar, no

garantizan fase lineal: el retardo que introduce el filtro varía con la frecuencia, lo que distorsiona la morfología temporal de la señal.

En segundo lugar, pueden ser inestables. La presencia de realimentación implica que si los parámetros del filtro no se diseñan o implementan correctamente, el sistema puede volverse inestable y producir una salida que crece sin límite. Por esta razón, la verificación de estabilidad es un paso obligatorio en todo diseño IIR.

4.4 Métodos de diseño

4.4.1 Diseño de filtros IIR — Transformada Bilineal

El método más utilizado para diseñar filtros IIR consiste en partir de un filtro analógico clásico y convertirlo al dominio digital mediante la Transformada Bilineal (TBL). La idea fundamental es aprovechar décadas de teoría de filtros analógicos, donde existen prototipos matemáticamente bien caracterizados, y trasladarlos al dominio discreto de forma sistemática.

La Transformada Bilineal establece una correspondencia matemática entre el dominio analógico y el dominio digital que preserva la estabilidad: un filtro analógico estable siempre produce un filtro digital estable al aplicar esta transformación. Sin embargo, introduce una distorsión no lineal del eje de frecuencias denominada warping o compresión frecuencial. Para compensarla, las frecuencias de corte deseadas deben pre-distorsionarse antes de diseñar el prototipo analógico, de modo que la distorsión introducida por la transformación las lleve exactamente a los valores digitales deseados.

4.4.2 Prototipos analógicos

Los prototipos analógicos son filtros matemáticamente idealizados que sirven como punto de partida para el diseño. Los más utilizados son cuatro. El filtro Butterworth presenta una respuesta en magnitud monótona en ambas bandas, sin rizado, lo que lo hace adecuado cuando se requiere una respuesta suave sin oscilaciones. El filtro Chebyshev tipo I introduce rizado controlado en la banda de paso a cambio de una transición más selectiva para el mismo orden. El filtro Chebyshev tipo II presenta rizado en la banda de rechazo manteniendo la banda de paso monótona. El filtro Elíptico presenta rizado en ambas bandas pero logra la transición más selectiva posible para un orden dado, siendo el más eficiente en términos de orden mínimo.

Para el presente trabajo se seleccionó el prototipo Butterworth debido a que las especificaciones del filtro son moderadas y a que la respuesta monótona sin rizado es conveniente para no introducir distorsiones en la señal fisiológica procesada.

4.4.3 Diseño de filtros FIR — Método de ventanas

El método de ventanas es el procedimiento más directo e intuitivo para diseñar filtros FIR. Parte de la respuesta al impulso de un filtro ideal, que tendría una transición perfectamente abrupta entre la banda de paso y la banda de rechazo, y la trunca multiplicándola por una función de ventana de longitud finita. Este truncamiento convierte la respuesta infinita del filtro ideal en una respuesta finita implementable.

La elección de la función de ventana determina el compromiso entre dos características contrapuestas: el ancho de la banda de transición y el nivel de rizado en las bandas. La ventana rectangular produce la transición más estrecha pero introduce el mayor rizado, con una atenuación máxima de aproximadamente 21 dB en la banda de rechazo. La ventana de Hamming reduce el rizado a 53 dB a costa de una banda de transición algo más ancha. La

ventana de Blackman alcanza atenuaciones de 74 dB pero con una transición considerablemente más ancha. La elección depende de las especificaciones del sistema: si la atenuación en la banda de rechazo es el parámetro crítico, se selecciona una ventana con mayor atenuación aunque ello implique un orden mayor.

4.5 Fase lineal

La fase lineal es una propiedad de ciertos filtros digitales por la cual todas las componentes frecuenciales de la señal son retardadas exactamente el mismo tiempo al atravesar el filtro. Esto equivale a decir que el filtro introduce un retardo puro, sin distorsión de forma de onda.

En términos prácticos, un filtro con fase lineal preserva intacta la morfología temporal de la señal: si la entrada es una forma de onda compleja, la salida es exactamente la misma forma de onda desplazada en el tiempo. Por el contrario, un filtro sin fase lineal retarda distintas frecuencias en distinta medida, deformando la forma de onda de salida respecto a la entrada.

Esta propiedad es especialmente importante en el procesamiento de señales fisiológicas. En el contexto del presente trabajo, la curva de glucosa en sangre contiene información clínica en su morfología temporal: la velocidad de ascenso o descenso, la duración de los episodios hipoglucémicos o hiperglucémicos y la tendencia de la señal son parámetros diagnósticos relevantes. Un filtro con distorsión de fase podría alterar estas características y conducir a interpretaciones incorrectas.

Los filtros FIR con coeficientes simétricos garantizan fase lineal estricta. Los filtros IIR no poseen esta propiedad en general, lo que constituye una de sus principales limitaciones en aplicaciones donde la fidelidad morfológica de la señal es crítica.

4.6 Estabilidad de filtros digitales

La estabilidad es una propiedad fundamental de todo sistema de procesamiento de señales. Un filtro digital es estable si ante cualquier señal de entrada acotada produce una señal de salida también acotada, condición conocida como estabilidad BIBO (Bounded Input, Bounded Output).

En el dominio de la transformada Z, la condición de estabilidad se verifica analizando la ubicación de los polos de la función de transferencia del filtro en el plano complejo z. El filtro es estable si y solo si todos sus polos se encuentran estrictamente dentro del círculo unitario, es decir, a una distancia menor que uno del origen.

Los filtros FIR son incondicionalmente estables porque sus únicos polos se encuentran en el origen del plano z, que siempre está dentro del círculo unitario. Los filtros IIR pueden ser inestables si algún polo se ubica fuera del círculo unitario, situación que puede ocurrir tanto por un diseño incorrecto como por errores de redondeo numérico en la implementación. Este riesgo es especialmente relevante cuando la frecuencia de corte del filtro es muy baja respecto a la frecuencia de muestreo, ya que en ese caso los polos se acercan peligrosamente al borde del círculo unitario y pequeños errores de precisión numérica pueden desplazarlos hacia afuera.

4.7 Estructuras de implementación

Una vez obtenidos los coeficientes de un filtro digital, existen distintas formas de organizar las operaciones aritméticas para calcular la salida, denominadas estructuras de implementación.

La elección de la estructura afecta la sensibilidad numérica, el consumo de memoria y la eficiencia computacional.

La Forma Directa I implementa la ecuación en diferencias de manera literal, utilizando registros separados para las muestras de entrada y de salida pasadas. Es la más intuitiva pero requiere la mayor cantidad de memoria.

La Forma Directa II comparte los registros entre la parte recursiva y no recursiva del filtro, reduciendo a la mitad los registros necesarios. Es la estructura canónica más utilizada en implementaciones prácticas.

La Forma Directa II Transpuesta es una variante de la anterior con mayor robustez numérica, recomendada especialmente para implementaciones en aritmética de punto fijo o flotante de precisión reducida, como es el caso de microcontroladores.

La cascada de Secciones de Segundo Orden, conocida como SOS, descompone el filtro en una cadena de secciones de orden 2 conectadas en serie. Cada sección tiene polos y ceros cercanos entre sí, lo que reduce drásticamente la sensibilidad a errores de redondeo. Es la estructura recomendada para filtros de orden elevado o cuando la estabilidad numérica es crítica.

4.8 Sistema de decisión — Agente basado en reglas

Un sistema experto basado en reglas es un mecanismo de inteligencia artificial que codifica conocimiento especializado de un dominio en forma de reglas explícitas del tipo condición-acción. Cuando se presenta un conjunto de condiciones de entrada, el sistema evalúa qué reglas se activan y produce una recomendación o acción correspondiente.

En el contexto del diseño de filtros digitales, el agente recibe como entrada un conjunto de restricciones y requerimientos del sistema, tales como la frecuencia de muestreo, la memoria disponible en el microcontrolador, el requerimiento de fase lineal, el nivel de ruido de la señal y las restricciones de latencia. A partir de estas variables, el agente evalúa las reglas codificadas y produce una recomendación fundamentada sobre el tipo de filtro más adecuado para ese escenario particular, junto con la estructura de implementación sugerida y una justificación técnica de la decisión.

La principal ventaja de este enfoque es su total interpretabilidad: cada recomendación puede trazarse directamente a las reglas que se activaron, lo que facilita la validación y la documentación. Las reglas se derivan directamente de los fundamentos teóricos desarrollados en las secciones anteriores, garantizando consistencia entre el marco teórico y el sistema de decisión implementado.

V_ Análisis del Problema y Definición de Especificaciones

5.1 Identificación y caracterización de la señal

Naturaleza física y origen de la señal

El presente proyecto aborda el procesamiento de una señal fisiológica proveniente de un Monitor Continuo de Glucosa (CGM simulado). Esta señal representa las variaciones de los niveles de glucosa en el tiempo, caracterizándose por ser una variable fisiológica de dinámica muy lenta.

La fotoplethysmografía es una técnica óptica no invasiva que detecta variaciones en el volumen sanguíneo periférico a través de cambios en la absorción de luz infrarroja. Estas variaciones están directamente asociadas al ciclo cardíaco y permiten estimar parámetros fisiológicos como la saturación periférica de oxígeno (SpO₂) y la frecuencia cardíaca.

Frecuencia de muestreo

Se adopta $f_s = 10 \text{ Hz}$ ($T_s = 100 \text{ ms}$). La componente fisiológica de interés se encuentra por debajo de 0,5 Hz. Aplicando el Teorema de Nyquist-Shannon:

$$f_s \geq 2 * f_{max} = 1 \text{ Hz}$$

Con $f_s = 10 \text{ Hz}$ se supera ampliamente el mínimo teórico (margen $\times 10$), garantizando ausencia de aliasing, resolución temporal suficiente para filtros FIR de orden elevado, y carga computacional trivial para el ESP32.

Componentes espectrales de la señal útil

La banda pasante útil es 0.01–0.5 Hz para variación lenta (tendencia y componentes respiratorias muy bajas hasta 0.3 Hz), con interés secundario en armónicos cardíacos hasta 1–2 Hz si se desean características de pulso.

Potencia concentrada en bajo rango de frecuencia; relación señal-ruido típica dependiente del sensor y A/D.

Componentes de ruido e interferencia

Deriva o tendencia del sensor (drift) y variación DC lenta: $< 0.01 \text{ Hz}$, causa desplazamientos de línea base.

Ruido de cuantización del ADC: espectro amplio (casi blanco), amplitud depende de resolución A/D.

Ruido de alta frecuencia (movimiento, EMI): bandas por encima de 1–5 Hz (artefactos por movimiento, interferencia de iluminación).

Posibles interferencias periódicas (50/60 Hz) por iluminación o mains, atenuables con anti-aliasing y filtrado analógico previo si necesario.

5.2 Especificaciones técnicas del sistema de filtrado

Para adecuar la señal, se implementará una cadena de procesamiento compuesta por dos etapas:

Tipo de filtros y frecuencias de corte:

Filtro Pasa Bajos FIR: Se utilizará para suavizar la señal y eliminar el ruido de cuantización. Su frecuencia de corte se establece en $f_c = 0.5 \text{ Hz}$.

Filtro Pasa Altos IIR (Sustracción de tendencia): Se aplicará un filtro de primer orden para eliminar la deriva DC del sensor, con una frecuencia de paso enfocada en 0.01 Hz

Justificación de la fase lineal: Es un requisito de diseño crítico que el filtro pasa bajos principal (FIR) posea fase lineal. En el procesamiento de señales fisiológicas, la distorsión de fase altera la morfología temporal de la onda debido a retardos variables en distintas frecuencias. El uso de un filtro FIR de fase lineal asegura un retardo de grupo constante, preservando intacta la forma de la curva de glucosa en el tiempo.

Tabla de Requerimientos Técnicos

A continuación, se tabulan las especificaciones de diseño:

Parámetro	Filtro 1 (Suavizado de Ruido)	Filtro 2 (Supresión de Deriva)
Tipo de Filtro	Pasa Bajos	Pasa Altos
Arquitectura	FIR	IIR
Frecuencia de Muestreo (f_s)	10 Hz	10 Hz
Frecuencia de Paso (f_p)	0.5 Hz	0.01 Hz
Frecuencia de Rechazo (f_r)	1.5 Hz	0.001 Hz
Atenuación en banda de rechazo (A_s)	>40 dB	>20 dB
Ondulación máxima (R_p)	<1 dB	<1 dB
Restricción de Fase	Fase Lineal Estricta	No crítica

VI_ Diseño de filtros

La etapa de filtrado se implementó mediante la cascada de un filtro FIR pasa bajos y un filtro IIR pasa altos. Esta configuración se eligió porque permite satisfacer los requerimientos del sistema con un buen compromiso entre la calidad del procesamiento, la carga computacional y la implementación en el ESP32.

En principio, ambas funciones podrían realizarse con un único filtro pasa banda. Sin embargo, esta alternativa no resulta la más conveniente. Un filtro FIR pasa banda con frecuencias de corte de 0,01 Hz y 0,5 Hz, considerando una frecuencia de muestreo de 10 Hz, requeriría un orden elevado debido a la estrecha banda de transición, lo que incrementaría la cantidad de coeficientes y el número de operaciones por muestra. En cambio, un filtro IIR alcanzaría una respuesta similar con un orden considerablemente menor, pero introduciría una respuesta de fase no lineal sobre la banda de interés, modificando la forma temporal de la señal.

Por este motivo, se optó por combinar ambos tipos de filtros. El filtro FIR se emplea en la etapa de suavizado para preservar la morfología de la señal gracias a su fase lineal, mientras que el filtro IIR se utiliza para eliminar la deriva de muy baja frecuencia con un costo computacional reducido. Dado que el IIR actúa únicamente sobre componentes que posteriormente son atenuados, la distorsión de fase introducida en esa región no afecta la información útil de la señal.

Con esta estrategia se obtiene una solución eficiente desde el punto de vista computacional, adecuada para su ejecución en el ESP32 y capaz de conservar las características de la señal relevantes para el análisis. A continuación, se desarrolla el diseño matemático de cada una de las etapas de filtrado, comenzando por el filtro FIR pasa bajos y posteriormente por el filtro IIR pasa altos.

Filtro FIR Pasa bajos

El diseño de filtros FIR mediante el método de ventanas consiste en truncar la respuesta al impulso ideal, que es de duración infinita, multiplicándola por una función de ventana de longitud finita. La elección de la ventana determina el compromiso entre el ancho de la banda de transición y la atenuación en la banda de rechazo.

Se selecciona la ventana de Kaiser por ser la única ventana estándar que permite controlar simultáneamente estos dos parámetros mediante dos variables independientes: la longitud $(M + 1)$ y el parámetro β . Las ventanas de forma fija como Rectangular, Hanning, Hamming o Blackman tienen atenuaciones predeterminadas que no pueden ajustarse sin modificar el orden. Kaiser descubrió empíricamente que utilizando la función de Bessel modificada de primera especie es posible construir una ventana cuasi-óptima que se adapta a cualquier especificación de diseño sin necesidad de iteraciones, lo que la convierte en la herramienta estándar para diseño de filtros FIR en ingeniería.

Para la especificación de $A_s \geq 40$ dB con una banda de transición de 1 Hz, esta ventana proporciona el menor orden posible, reduciendo directamente el costo computacional en el ESP32.

Especificaciones de diseño

Parámetro	Símbolo	Valor
Frecuencia de muestreo	f_s	10 Hz
Frecuencia de paso	f_p	0,5 Hz
Frecuencia de rechazo	f_r	1,5 Hz
Atenuación mínima en rechazo	A_s	40 dB
Ondulación máxima en paso	R_p	1 dB
Fase	—	Lineal obligatoria

Cálculo del orden del filtro

Las ecuaciones de diseño de filtros FIR se expresan en función de la frecuencia angular digital. Por este motivo, el primer paso consiste en convertir las frecuencias de paso y rechazo desde Hertz a radianes por muestra utilizando la frecuencia de muestreo del sistema.

La frecuencia angular correspondiente al borde de la banda es la siguiente:

$$\omega_p = \frac{f_p}{f_s} \cdot 2\pi = \frac{0,5}{10} \cdot 2\pi = 0,1\pi \text{ rad/muestra}$$

Análogamente, la frecuencia angular correspondiente al inicio de la banda de rechazo resulta:

$$\omega_r = \frac{f_r}{f_s} \cdot 2\pi = \frac{1,5}{10} \cdot 2\pi = 0,3\pi \text{ rad/muestra}$$

Una vez determinadas ambas frecuencias, se calcula el ancho de la banda de transición, parámetro que determina el orden mínimo requerido por el filtro.

$$\Delta\omega = \omega_r - \omega_p = 0,3\pi - 0,1\pi = 0,2\pi \text{ rad/muestra}$$

Cuanto menor sea el ancho de esta banda de transición, mayor será el orden necesario para aproximar la respuesta ideal del filtro.

Determinación del parámetro β

La ventana de Kaiser incorpora un parámetro denominado β , responsable de controlar la forma de la ventana. Este parámetro establece el compromiso existente entre el nivel de atenuación de los lóbulos laterales y el ancho del lóbulo principal de la respuesta espectral.

Para una atenuación comprendida entre 21 dB y 50 dB, β se determina mediante la siguiente expresión:

$$\beta = 0,5842 (A_s - 21)^{0,4} + 0,07886 (A_s - 21)$$

Reemplazando el valor de atenuación por 40 dB

$$\beta = 0,5842 (19)^{0,4} + 0,07886 (19) = 1,9833 + 1,4983 = 3,3953$$

Determinación de orden M

Una de las principales ventajas de la ventana de Kaiser es que permite estimar directamente el orden mínimo requerido para satisfacer simultáneamente la atenuación y el ancho de transición especificados.

El orden aproximado se obtiene mediante:

$$M = \frac{A_s - 8}{2,285 \cdot \Delta\omega}$$

Sustituyendo los valores obtenidos anteriormente:

$$M = \frac{40 - 8}{2,285 \cdot 0,2\pi} = \frac{32}{1,4354} = 22,3$$

Esta fórmula tiene una precisión de ± 2 , por lo que se adopta $M=23$ tomando el entero impar superior. Un valor impar de M garantiza un FIR Tipo I, que posee simetría exacta y un número entero de coeficientes a cada lado del centro, condición necesaria para la fase lineal perfecta. La longitud total del filtro es:

$$L = M + 1 = 24 \text{ coeficientes}$$

Cálculo analítico de los coeficientes

Una vez determinado el orden del filtro, se calcula la respuesta impulsiva del filtro pasa bajos ideal. Para ello se adopta una frecuencia de corte ubicada en el punto medio de la banda de transición, con el objetivo de repartir de manera uniforme el error de aproximación entre la banda pasante y la banda de rechazo.

$$\omega_c = \frac{\omega_p + \omega_r}{2} = \frac{0,1\pi + 0,3\pi}{2} = 0,2\pi \text{ rad/muestra}$$

La respuesta al impulso del filtro pasa bajos ideal, causal y centrada en $\alpha=M/2=11,5$, es:

$$h_d[n] = \frac{\sin[\omega_c(n - \alpha)]}{\pi(n - \alpha)}, \quad n \neq \alpha, \quad 0 \leq n \leq M$$

Dado que $M = 23$ es impar, $\alpha = 11,5$ no es entero y no existe muestra central. Esto es consistente con un FIR Tipo I de longitud par, donde la simetría se cumple en todos los coeficientes sin excepción. La ventana de Kaiser, definida a partir de la función de Bessel modificada $I_0(\cdot)$, es:

$$w[n] = \frac{I_0 \left(\beta \sqrt{1 - \left(\frac{n - \alpha}{\alpha} \right)^2} \right)}{I_0(\beta)}, \quad 0 \leq n \leq M$$

donde I_0 representa la función de Bessel modificada de primera especie y orden cero. Al multiplicar la respuesta ideal por la ventana se obtienen los coeficientes finales del filtro:

$$h[n] = h_d[n] \cdot w[n], \quad n = 0, 1, \dots, 23$$

Los valores numéricos fueron calculados en MATLAB mediante las ecuaciones anteriores con. Se presentan los primeros y últimos cinco coeficientes con seis cifras significativas

$$\begin{aligned} h[0] &= h[23] = 0,003345 \\ h[1] &= h[22] = 0,002155 \\ h[2] &= h[21] = -0,003322 \\ h[3] &= h[20] = -0,012684 \\ h[4] &= h[19] = -0,022079 \end{aligned}$$

Verificación de simetría:

$$h[n] = h[M - n], \quad \forall n = 0, 1, \dots, 23$$

Error numérico de simetría:

$$\max |h[n] - h[M - n]| \approx 10^{-17} \approx 0$$

Verificación de simetría y fase lineal

$$h[n] = h[M - n], \quad \forall n = 0, 1, \dots, M$$

Esta simetría tiene una consecuencia directa en el dominio de la frecuencia. La respuesta en frecuencia adopta la forma:

$$H(e^{j\omega}) = A(\omega) e^{-j\omega\alpha}$$

donde $A(\omega)$ es una función puramente real. La fase del sistema es entonces

$$\phi(\omega) = -\alpha \omega = -\frac{M}{2} \omega = -11,5 \omega$$

La pendiente de la fase es constante e igual a $\frac{-M}{2}$ para toda frecuencia. Esto implica un retardo de grupo uniforme:

$$\tau_g = -\frac{d\phi}{d\omega} = \frac{M}{2} = 11,5 \text{ muestras} = 1,15 \text{ s}$$

En la práctica, esto significa que todas las componentes espectrales de la señal de glucosa son retardadas exactamente al mismo tiempo, preservando su morfología sin distorsión. Esta propiedad es fundamental en aplicaciones de monitoreo fisiológico donde la forma de la señal contiene información clínica relevante.

Función de transferencia

La función de transferencia del filtro FIR se obtiene aplicando la transformada Z a la ecuación de convolución:

$$H(z) = \sum_{n=0}^M h[n] z^{-n}$$

Expandiendo con los coeficientes calculados:

$$H(z) = h[0] + h[1] z^{-1} + h[2] z^{-2} + \dots + h[23] z^{-23}$$

Aprovechando la simetría de los coeficientes, la expresión puede compactarse agrupando términos simétricos

$$H(z) = \sum_{n=0}^{11} h[n] (z^{-n} + z^{-(23-n)})$$

Esta forma demuestra que el filtro requiere solo 12 multiplicaciones únicas por muestra en lugar de 24, lo que reduce a la mitad el costo computacional en la implementación sobre el ESP32.

El filtro FIR es un sistema sin retroalimentación: la salida depende únicamente de muestras presentes y pasadas de la entrada. En consecuencia, todos sus polos se ubican en el origen del plano z , garantizando estabilidad incondicional independientemente de los valores de los coeficientes. Esta es una ventaja fundamental respecto a los filtros IIR, que analizamos en la siguiente sección.

Filtro IIR Pasa altos

Para el diseño del filtro IIR se debe elegir un prototipo analógico clásico como punto de partida. Los prototipos disponibles son Butterworth, Chebyshev I, Chebyshev II y Elíptico, cada uno con distintas características de respuesta en magnitud y fase.

Se selecciona el prototipo **Butterworth** por las siguientes razones: las especificaciones del filtro son moderadas ($A_s = 20$ dB, $R_p = 1$ dB), por lo que no se requiere alta selectividad. Butterworth presenta respuesta monótona en ambas bandas sin rizado, lo que evita distorsión en la señal fisiológica. Además, su implementación con la Transformada Bilineal es algebraicamente simple y produce coeficientes numéricamente estables para órdenes bajos.

Cálculo del orden mínimo N

Dado que la Transformada Bilineal introduce una distorsión no lineal del eje de frecuencias, las frecuencias digitales de corte deben convertirse a sus equivalentes analógicos mediante el

proceso de pre-distorsión antes de calcular el orden del filtro:

$$\Omega_p = \tan \left(\frac{\pi \cdot f_p}{f_s} \right) = \tan \left(\frac{\pi \cdot 0.01}{10} \right) \approx 0.003142 \text{ rad/s}$$

$$\Omega_s = \tan \left(\frac{\pi \cdot f_r}{f_s} \right) = \tan \left(\frac{\pi \cdot 0.001}{10} \right) \approx 0.0003142 \text{ rad/s}$$

Dado que el filtro es pasa-altos, la banda de rechazo se encuentra en frecuencias menores que fp. El cociente de selectividad resulta:

$$\frac{\Omega_p}{\Omega_s} = \frac{0.003142}{0.0003142} \approx 10$$

El orden mínimo de un filtro Butterworth que satisface simultáneamente las restricciones de banda de paso y banda de rechazo se obtiene mediante:

$$N \geq \frac{\log \left(\frac{10^{A_s/10} - 1}{10^{R_p/10} - 1} \right)}{2 \log \left(\frac{\Omega_p}{\Omega_s} \right)}$$

Sustituyendo As = 20 dB, Rp = 1 dB y Ωp/Ωs = 10:

$$N \geq \frac{\log \left(\frac{10^2 - 1}{10^{0.1} - 1} \right)}{2 \log(10)} = \frac{\log \left(\frac{99}{0.2589} \right)}{2} = \frac{\log(382.4)}{2} = \frac{2.5826}{2} \approx 1.29$$

Redondeando al entero superior inmediato:

$$\boxed{N = 2}$$

Función de transferencia analógica Ha(s)

El prototipo pasa-bajos normalizado Butterworth de orden 2 tiene la siguiente función de transferencia estándar, con frecuencia de corte unitaria (Ωc = 1 rad/s):

$$H_{LP,norm}(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

Los polos del prototipo se ubican en el semiplano izquierdo del plano s, garantizando estabilidad analógica:

$$s_{1,2} = -\frac{\sqrt{2}}{2} \pm j \frac{\sqrt{2}}{2} \approx -0.7071 \pm j 0.7071$$

Para convertir el prototipo pasa-bajos en pasa-altos con frecuencia de corte Ω_p , se aplica la transformación de variable:

$$s \longrightarrow \frac{\Omega_p}{s}$$

Sustituyendo en $H_{LP,norm}(s)$ y multiplicando numerador y denominador por s^2 :

$$H_a(s) = \frac{s^2}{s^2 + \sqrt{2} \Omega_p s + \Omega_p^2}$$

Reemplazando $\Omega_p = 0.003142$ rad/s:

$$H_a(s) = \frac{s^2}{s^2 + 0.004443 s + 9.872 \times 10^{-6}}$$

Los ceros de $H_a(s)$ se encuentran en $s = 0$ (doble), lo que confirma el comportamiento pasa-altos: la ganancia tiende a cero a baja frecuencia. Los polos se calculan resolviendo el denominador:

$$s_{1,2} = \frac{-0.004443 \pm \sqrt{(0.004443)^2 - 4 \cdot 9.872 \times 10^{-6}}}{2} = -0.002221 \pm j 0.002221$$

Todos los polos tienen parte real negativa ($\text{Re}(s) < 0$), por lo tanto $H_a(s)$ es analógicamente estable.

Aplicación de la Transformada Bilineal

La Transformada Bilineal (TBL) convierte $H_a(s)$ en $H(z)$ mediante la sustitución:

$$s = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} = 2f_s \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$

Con $f_s = 10$ Hz el factor vale $k = 2f_s = 20$. Se define $S = 20 \cdot (1 - z^{-1}) / (1 + z^{-1})$ para simplificar la notación. Calculando S^2 :

$$S^2 = \frac{400 (1 - 2z^{-1} + z^{-2})}{(1 + z^{-1})^2}$$

Sustituyendo en $H_a(s)$ y multiplicando numerador y denominador por $(1 + z^{-1})^2$, se expande

cada término del denominador por separado:

$$400 (1 - 2z^{-1} + z^{-2}) \rightarrow \text{de } S^2 \cdot (1 + z^{-1})^2$$

$$0.004443 \cdot 20 (1 - z^{-1})(1 + z^{-1}) = 0.08886 (1 - z^{-2}) \rightarrow \text{de } \sqrt{2} \Omega_p \cdot S \cdot (1 + z^{-1})^2$$

$$9.872 \times 10^{-6} (1 + z^{-1})^2 \approx 0 \rightarrow \text{de } \Omega_p^2 \cdot (1 + z^{-1})^2 \text{ (despreciable)}$$

Agrupando por potencias de z^{-1} se obtiene el denominador de $H(z)$:

$$\text{Den}(z) = (400 + 0.08886) - 800 z^{-1} + (400 - 0.08886) z^{-2} = 400.089 - 800 z^{-1} + 399.911 z^{-2}$$

Normalizando por el coeficiente de z^0 ($= 400.089$), la función de transferencia digital resulta:

$$H(z) = \frac{0.99557 (1 - 2z^{-1} + z^{-2})}{1 - 1.99111 z^{-1} + 0.99115 z^{-2}}$$

Ecuación en diferencias

Expresando $H(z) = Y(z)/X(z)$ y despejando algebraicamente:

$$Y(z) (1 - 1.9996 z^{-1} + 0.9996 z^{-2}) = 0.9998 X(z) (1 - 2z^{-1} + z^{-2})$$

Aplicando la propiedad de retardo ($z^{-1} \leftrightarrow$ desplazamiento de una muestra) y despejando $y[n]$:

$$y[n] = 0.99557 x[n] - 1.99113 x[n-1] + 0.99557 x[n-2] + 1.99111 y[n-1] - 0.99115 y[n-2]$$

Verificación de estabilidad

Un filtro IIR es estable si y sólo si todos sus polos se encuentran dentro del círculo unitario en el plano z ($|z| < 1$). Los polos de $H(z)$ son las raíces del denominador, equivalente a resolver:

$$z^2 - 1.9996 z + 0.9996 = 0$$

Aplicando la fórmula cuadrática:

$$z_{1,2} = \frac{1.9996 \pm \sqrt{(1.9996)^2 - 4 \cdot 0.9996}}{2} = \frac{1.9996 \pm \sqrt{1.6 \times 10^{-4}}}{2} = \frac{1.9996 \pm 0.01265}{2}$$

$$z_{1,2} = 0.995557 \pm j 0.004423$$

Verificando la condición de estabilidad para cada polo:

$$|z_{1,2}| = 0.99557 < 1 \Rightarrow \text{ambos polos dentro del círculo unitario} \checkmark$$

Los ceros se ubican en $z = 1$ (doble), lo que confirma atenuación total en frecuencia cero (DC). Ambos polos están dentro del círculo unitario, por lo tanto el filtro es estable.

VII_ Simulación de filtros MATLAB

FIR Pasa Bajo Simulado

Parámetros obtenidos

La función **kaiserord** calculó automáticamente el orden mínimo necesario para cumplir las especificaciones, operando sobre las desviaciones lineales equivalentes. El orden resultante fue $M = 24$, con longitud total $L = 25$ coeficientes. Este valor es consistente con el estimado analíticamente en la Sección 5 ($M = 23$), dentro del margen de precisión de ± 2 indicado por la fórmula de Kaiser. El retardo de grupo resultante es:

$$\tau_g = \frac{M}{2} = \frac{24}{2} = 12 \text{ muestras} = 1,2 \text{ s}$$

En la figura 1 se observa que la magnitud confirma el cumplimiento de todas las especificaciones de diseño. En la banda pasante, de 0 a 0,5 Hz, la atenuación es inferior a 1 dB. A partir de 1,5 Hz la atenuación supera los 40 dB, satisfaciendo el requisito de rechazo. La banda de transición entre ambas frecuencias presenta la pendiente esperada para el orden calculado. Los lóbulos en la banda de rechazo se mantienen por debajo de la línea de -40 dB en toda la banda, evidenciando el correcto control de lóbulos laterales que ofrece la ventana Kaiser mediante el parámetro β

La fase muestra un comportamiento estrictamente lineal en toda la banda de interés, con pendiente constante, lo que confirma la propiedad de fase lineal del filtro FIR Tipo I diseñado.

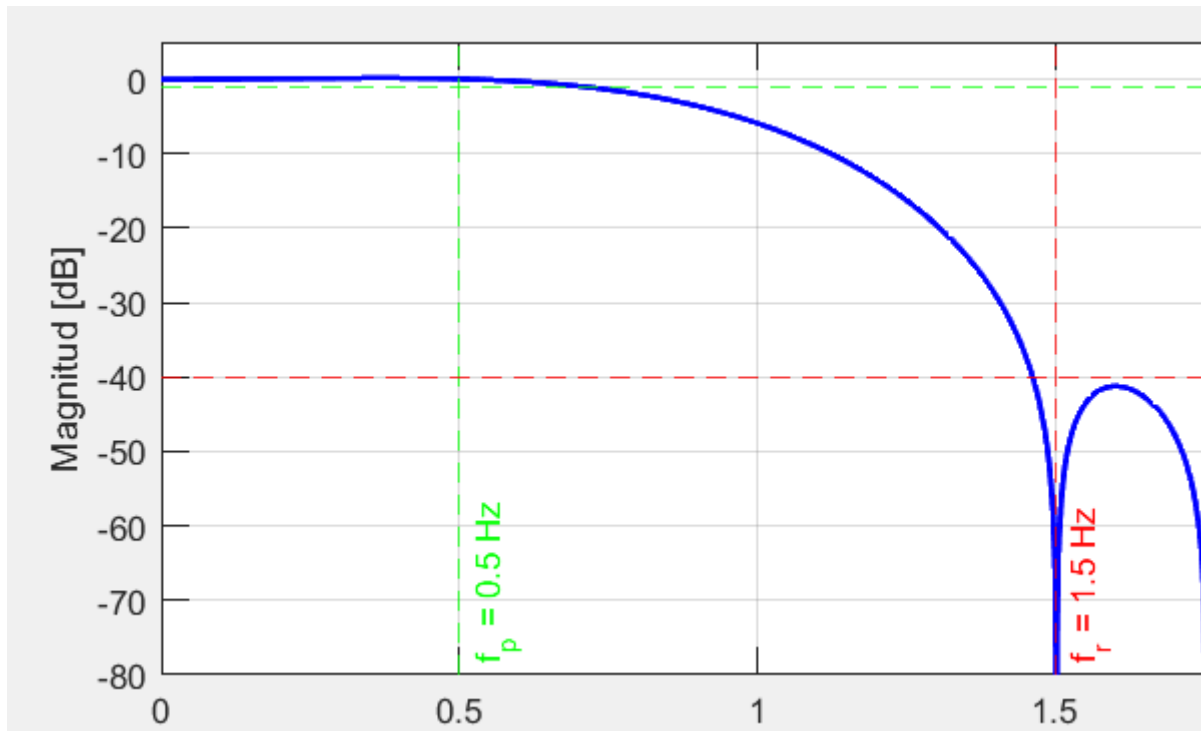


Fig 1. Respuesta en frecuencia del filtro Kaiser FIR con interés en frecuencias bajas

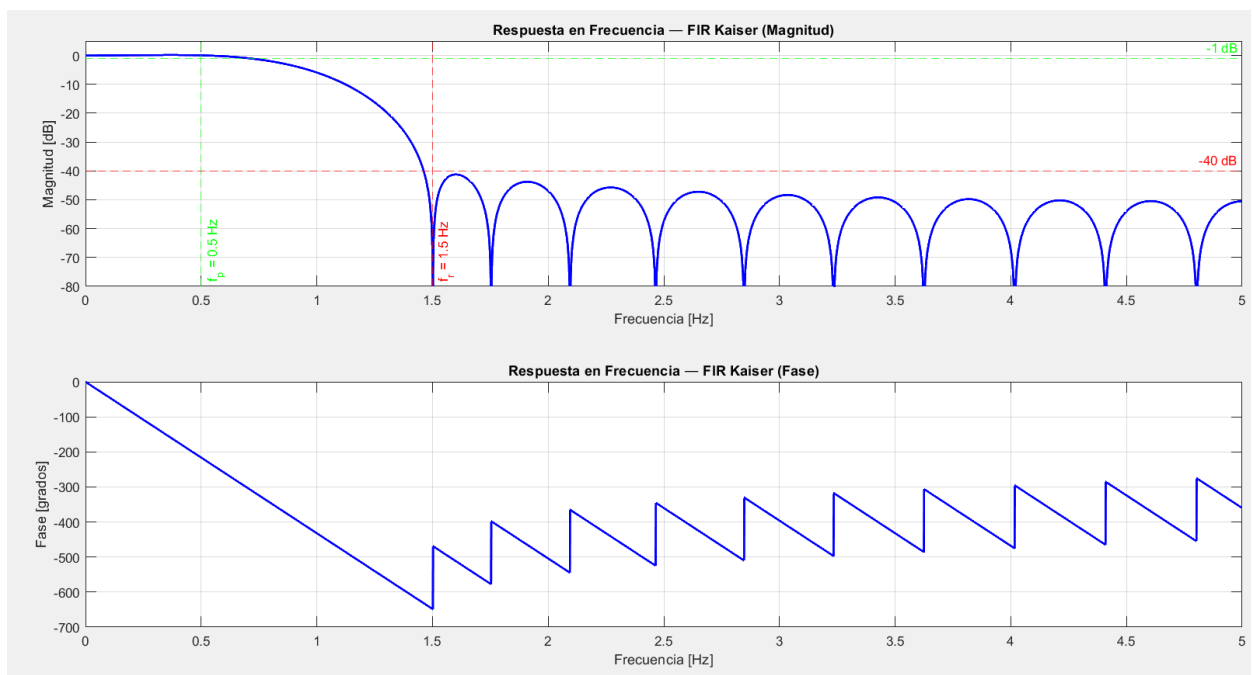


Fig 2. Respuesta en frecuencia Filtro Kaiser FIR

La respuesta al impulso muestra la distribución de los 25 coeficientes con simetría perfecta alrededor del centro $\alpha=M/2=12$.

Los coeficientes de mayor amplitud se concentran en la región central, aproximadamente entre las muestras 10 y 14, y decaen suavemente hacia los extremos.

Este comportamiento es característico de la ventana Kaiser: el suavizado progresivo hacia los bordes reduce la amplitud de los lóbulos laterales en el dominio de la frecuencia, a diferencia de la ventana rectangular que trunca abruptamente y genera el fenómeno de Gibbs.

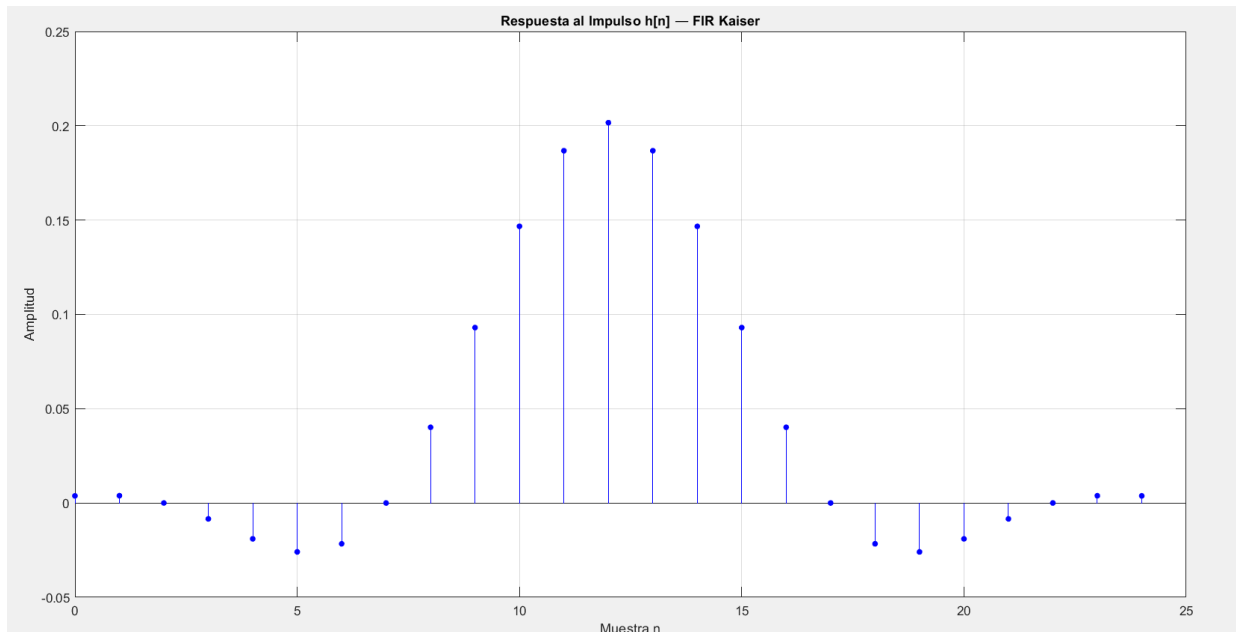


Fig 3. Respuesta al impulso $h[n]$

El retardo de grupo calculado con `grpdelay` es perfectamente constante e igual a 12 muestras para todas las frecuencias, desde 0 Hz hasta la frecuencia de Nyquist de 5 Hz. Esta plenitud absoluta válida cuantitativamente la fase lineal del filtro: todas las componentes espectrales de la señal de glucosa sufren exactamente el mismo retardo temporal de 1,2 segundos, sin distorsión de forma. Esta es la diferencia fundamental respecto al filtro IIR, que presenta retardo de grupo variable con la frecuencia, como se analizará en la sección de comparación.

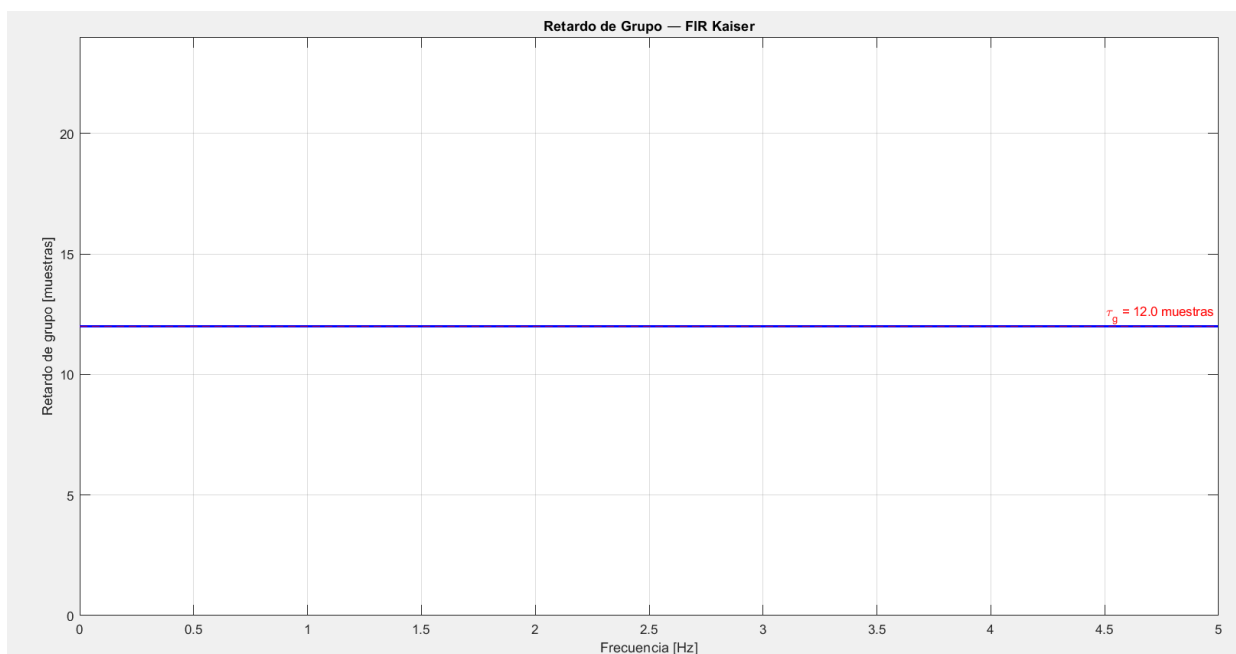


Fig 4. Retardo de grupo FIR

El diagrama polo-cero presenta 24 ceros, correspondientes al grado del polinomio del numerador, y un polo de multiplicidad 24 en el origen del plano z . Los ceros ubicados sobre o próximos al círculo unitario en la región de la banda de rechazo son responsables de la elevada atenuación en esas frecuencias. La ubicación de todos los polos en el origen confirma que el filtro FIR es BIBO estable. Además, pueden observarse algunos ceros fuera

del círculo unitario, lo cual es normal en filtros FIR de fase lineal debido a la simetría de sus coeficientes y no afecta la estabilidad del sistema.

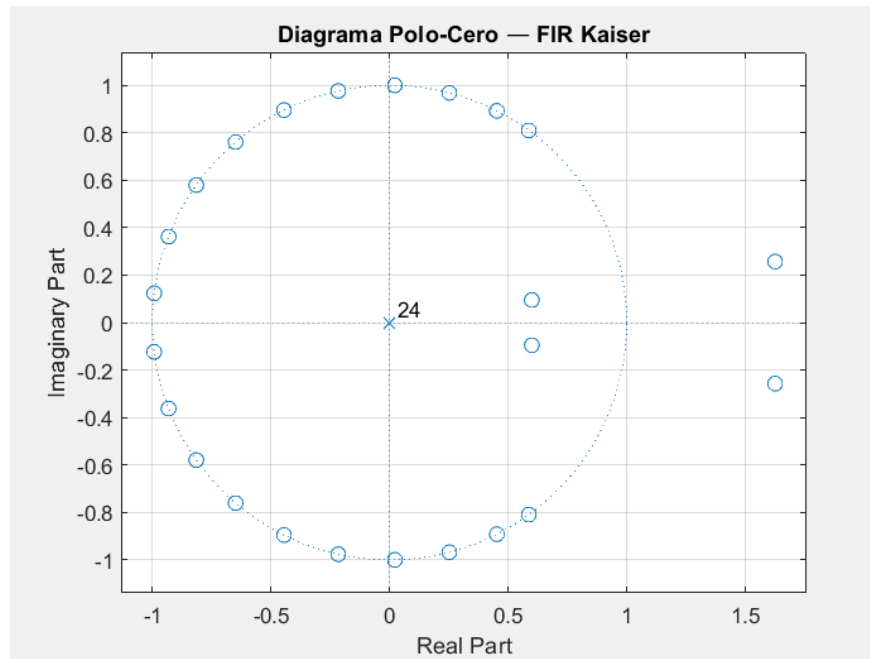


Fig 5. Diagrama de Polo-Cero

La señal de entrada fue construida como la superposición de tres componentes: una variación fisiológica lenta a 0,05 Hz representando la glucosa intersticial, una deriva lineal de línea de base simulando el drift del sensor, y ruido de alta frecuencia a 2 Hz más ruido blanco gaussiano. En el panel superior se observa la señal contaminada junto a la referencia sin ruido HF, donde el ruido es claramente visible como fluctuaciones rápidas sobre la tendencia lenta.

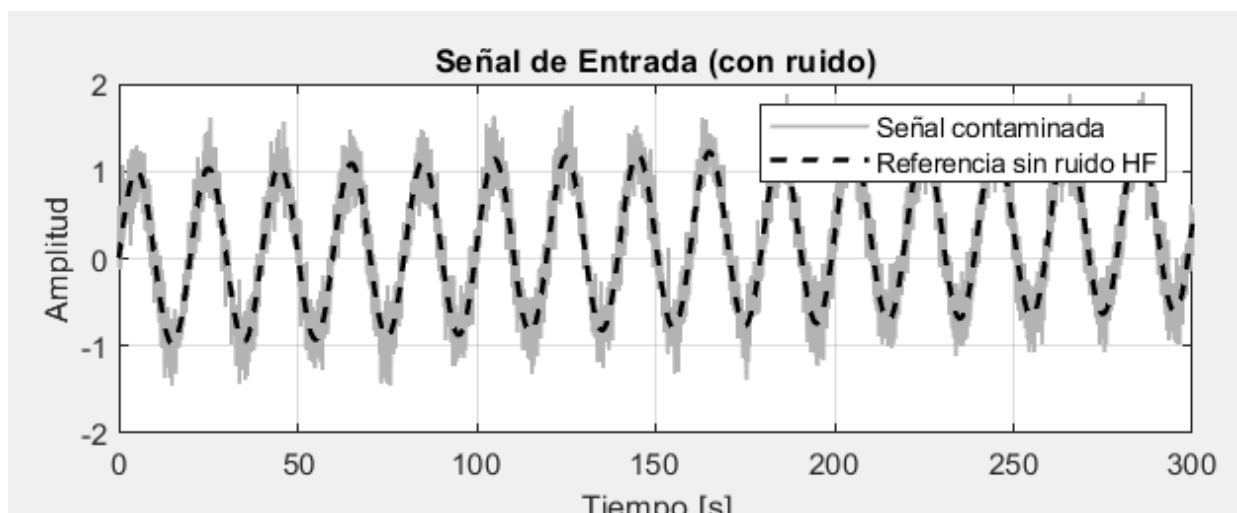


Fig 6. Señal de prueba con ruido

En la figura 7, la salida del filtro FIR reproduce fielmente la forma de la referencia. La componente de alta frecuencia fue eliminada y la morfología de la señal útil se preservó sin distorsión, gracias a la fase lineal del filtro. Se aprecia el retraso de 1,2 segundos entre la entrada y la salida, esperado para un filtro de orden 24.

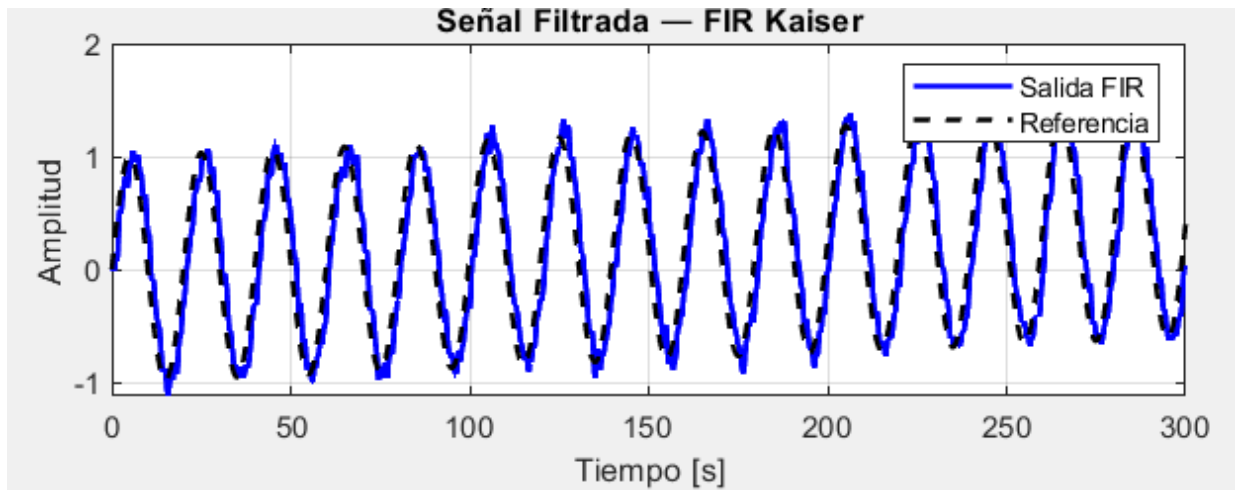


Fig 7. Salida filtrada de la señal con ruido

El espectro de la figura 8 confirma en el dominio de la frecuencia lo observado en el tiempo. La señal original presenta energía concentrada en baja frecuencia correspondiente a la componente fisiológica, y componentes visibles alrededor de 2 Hz correspondientes al ruido HF. La señal filtrada muestra una atenuación superior a 40 dB en la banda de rechazo a partir de 1,5 Hz, con la componente útil preservada por debajo de 0,5 Hz. La diferencia entre ambos espectros en la zona de rechazo evidencia la eficacia del filtro diseñado.

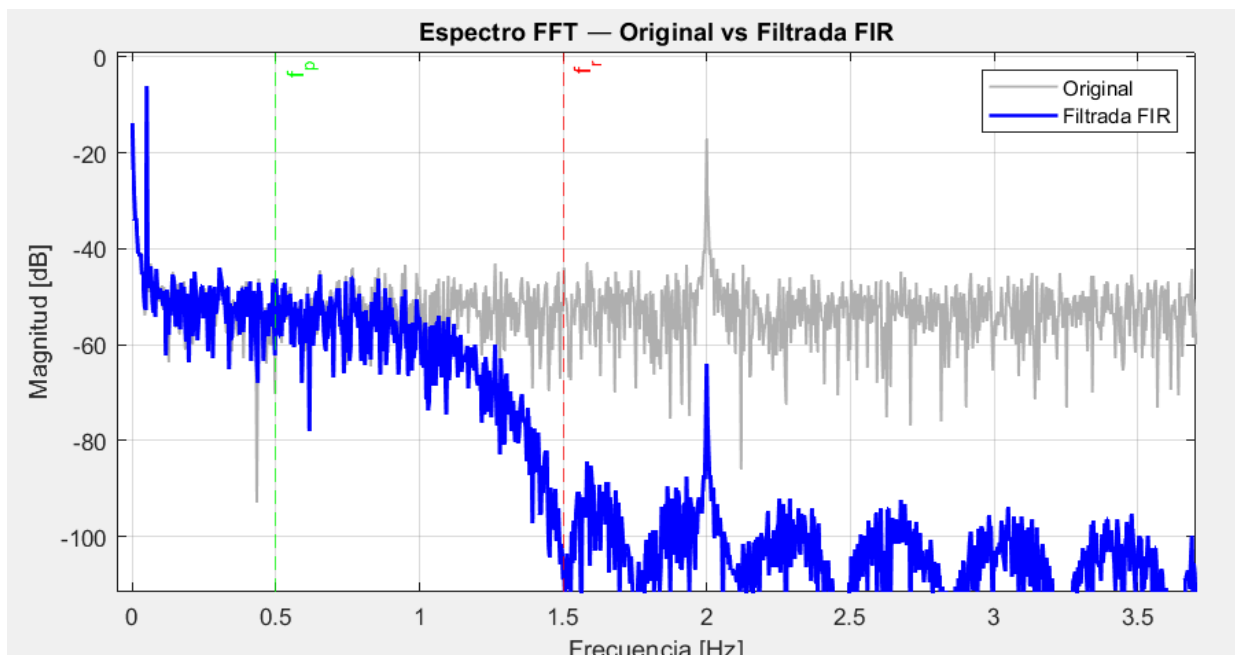


Fig 8. Espectro FFT de la señal original comparada con la señal filtrada

IIR Pasa Alto Simulado

Se realizó la implementación del filtro Butterworth, diseñándolo a partir de su plantilla de valores. Debido a esto pudimos comprobar de forma realista tanto cada valor de coeficientes como de ceros y polos.

Los coeficientes obtenidos analíticamente presentan leves diferencias respecto a los calculados por MATLAB debido al redondeo acumulado en los pasos intermedios de la TBL.

Para la implementación embebida se adoptan los coeficientes de MATLAB por su mayor precisión numérica.

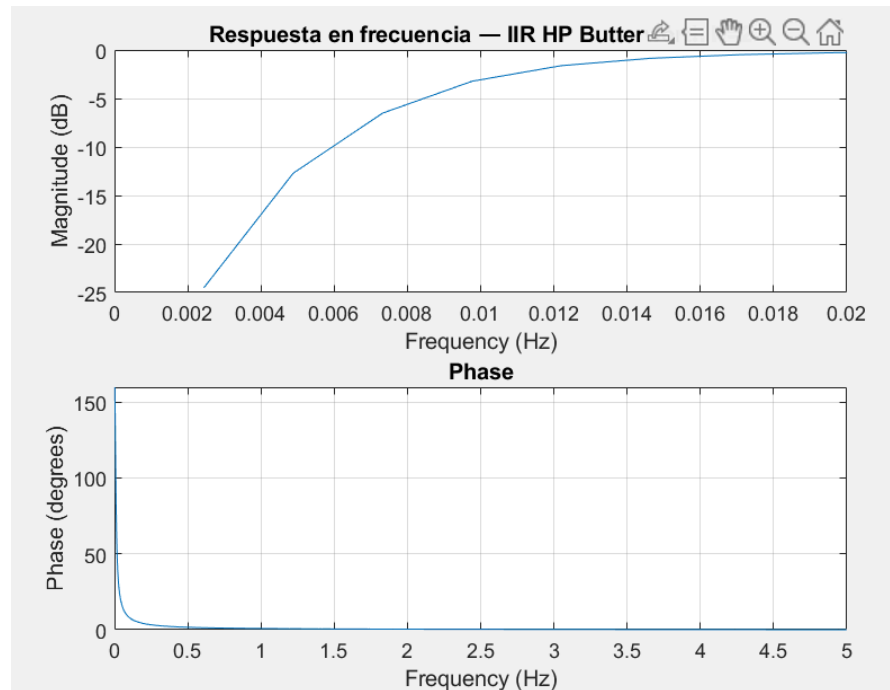


Fig 9. Respuesta en frecuencia del filtro IIR

Ubicación polos y zeros filtro FIR

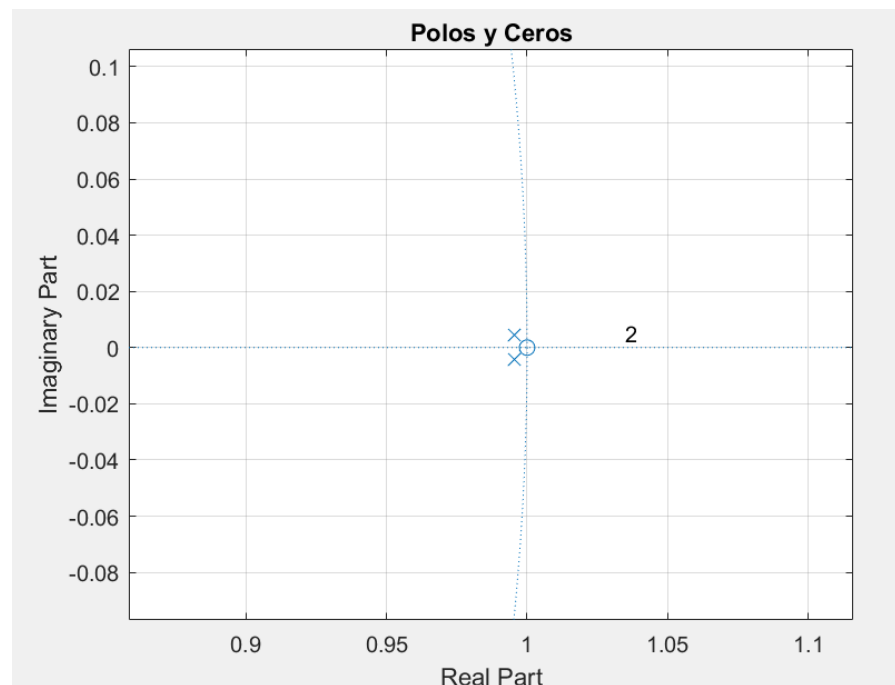


Fig 10. Posición de ceros y polos dentro del círculo unitario.

Respuesta al impulso IIR

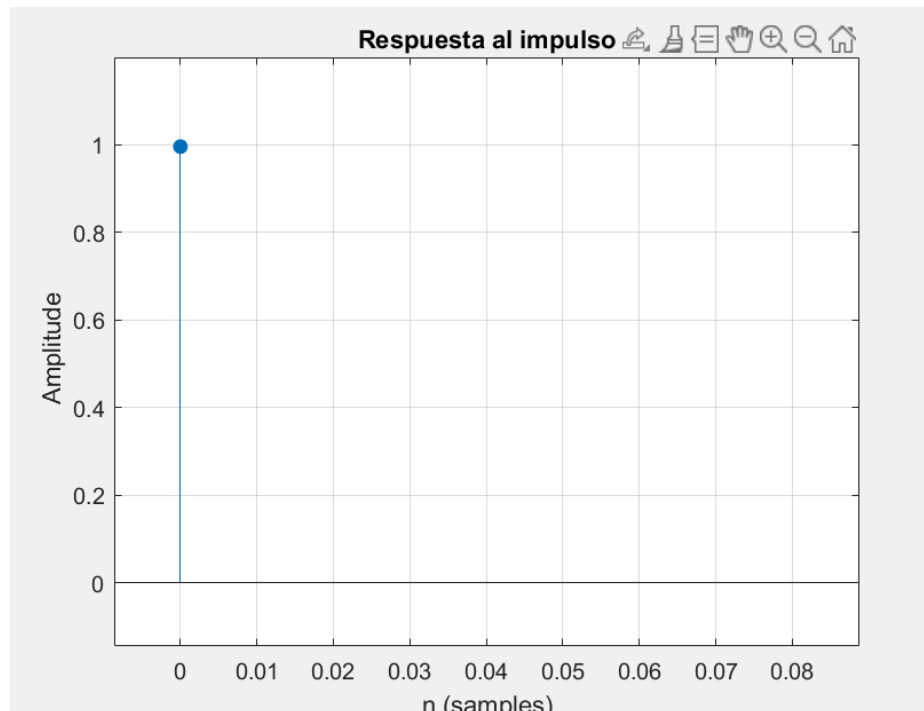


Fig 11. Respuesta al impulso filtro IIR

Retardo de grupo

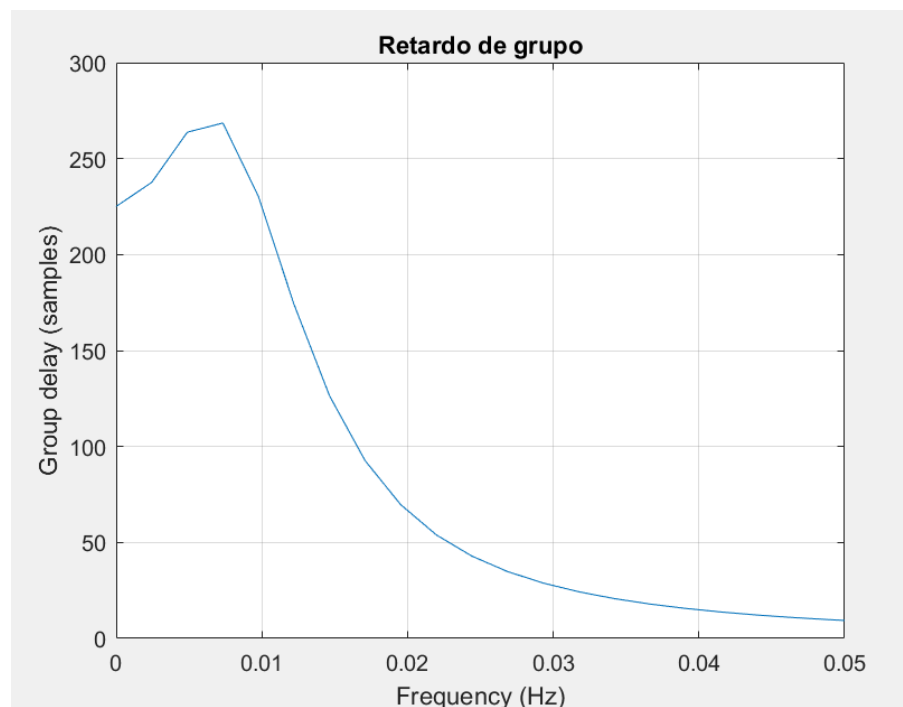


Fig 12. Retardo de grupo filtro IIR

La figura 12 muestra el retardo de grupo del filtro IIR Butterworth pasa-altos de orden 2. Se observa que el retardo de grupo no es constante, sino que depende de la frecuencia de la señal. El máximo retardo se presenta en la zona cercana a la frecuencia de corte (≈ 0.01 Hz), donde alcanza aproximadamente 260 muestras. A medida que aumenta la frecuencia, el

retardo disminuye progresivamente hasta valores inferiores a 10 muestras para frecuencias mayores a 0.05 Hz. Este comportamiento es característico de los filtros IIR de fase no lineal. Las componentes espectrales cercanas a la frecuencia de corte experimentan un retardo considerablemente mayor que las componentes de alta frecuencia. Como consecuencia, distintas frecuencias atraviesan el filtro con diferentes retardos temporales, produciendo una distorsión de fase o deformación de la forma de onda.

En contraste, un filtro FIR de fase lineal presenta un retardo de grupo prácticamente constante en toda la banda pasante, preservando la morfología temporal de la señal. Esta es una de las principales ventajas de los filtros FIR en aplicaciones biomédicas donde la forma de la señal posee información clínica relevante.

Señal simulada

Para validar el desempeño del filtro se generó una señal sintética representativa de un monitor continuo de glucosa (CGM). La señal útil se modeló mediante una componente senoidal de baja frecuencia (0.005 Hz), representando la variación fisiológica de la glucosa. Sobre esta señal se agregó una deriva lenta del sensor (offset de 100 mg/dL con pendiente de 0.05 mg/dL/s) y ruido blanco gaussiano de amplitud 0.5 mg/dL para simular el ruido de medición. La señal de entrada queda definida como:

$$x(t) = \text{señal útil} + \text{deriva} + \text{ruido}$$

```
%% FIGURA 3 -- Señal glucosa
t = 0:1/fs:600;
% Componentes de la señal
util = 20*sin(2*pi*0.005*t);           % variación lenta de glucosa (centrada
en 0)
deriva = 100 + 0.05*t;                 % nivel basal + drift lento
ruido = 0.5*randn(size(t));
x = util + deriva + ruido;             % señal completa con deriva
y = filter(b, a, x);                  % salida HP: elimina deriva, deja
variación
```

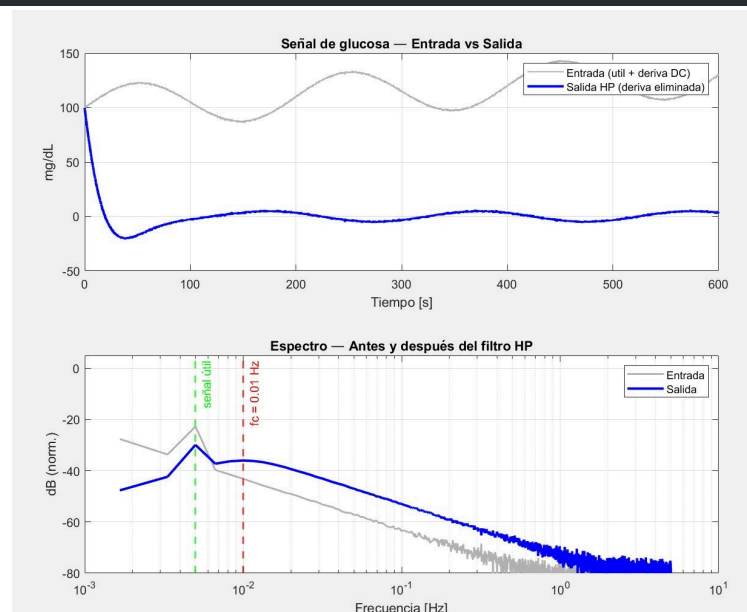


Fig 13. Verificación de filtrado con una señal simulada.

La Figura 13 muestra la señal antes y después del filtrado. Se observa que el filtro elimina eficazmente la componente continua y la deriva lenta del sensor, centrando la señal alrededor de cero. La componente de baja frecuencia correspondiente a la variación fisiológica permanece presente, mientras que el nivel DC es prácticamente eliminado. Se observa además un transitorio inicial, característico de los filtros IIR al comenzar el procesamiento.

Esta señal fue utilizada como entrada del filtro IIR pasa-altos diseñado.

La atenuación promedio obtenida en la banda de rechazo fue de aproximadamente **29.6 dB**, superando ampliamente la especificación mínima de 20 dB establecida para el filtro.

Métricas IIR

<u>Métrica</u>	<u>Valor</u>
SNR entrada	-18.23 dB
SNR salida	-3.15 dB
Mejora SNR	15.07 dB
RMSE	20.33 mg/dL

Comparaciones Filtros

Cabe destacar que los filtros diseñados poseen funciones diferentes. El filtro IIR corresponde a un pasa-altos destinado a eliminar la deriva y el nivel DC del sensor, mientras que el filtro FIR es un pasa-bajos orientado a atenuar el ruido de alta frecuencia. En consecuencia, las métricas de calidad (SNR, RMSE o atenuación) no son directamente comparables entre ambos filtros, ya que fueron evaluados sobre señales contaminadas con perturbaciones diferentes y persiguen objetivos distintos.

La comparación realizada se centra principalmente en las características estructurales de cada implementación, como el orden, costo computacional, fase, retardo de grupo, estabilidad y memoria requerida.

Aspecto	FIR Kaiser (PB)	IIR Butterworth (PA)
Función	Elimina ruido de alta frecuencia	Elimina deriva y componente DC
Orden	Alto	2
Costo computacional	Alto	Bajo
Fase	Lineal	No lineal
Retardo de grupo	Constante	Variable

Estabilidad	Siempre estable	Estable (polos dentro del círculo unitario)
Memoria	Mayor	Menor
Aplicación	Preservar la forma de onda	Corregir deriva del sensor

VIII_ Agentes de IA

Más allá de cumplir con el requisito mínimo de recomendar FIR o IIR ante un conjunto fijo de restricciones, se desarrolló una herramienta de software con interfaz gráfica en MATLAB que aborda el problema de selección de filtro digital como lo enfrentaría un ingeniero en la práctica: partiendo del conocimiento de dominio sobre el tipo de señal a procesar, y combinándolo dinámicamente con las restricciones reales de la plataforma de hardware disponible.

El agente incorpora las diez aplicaciones biomédicas e industriales especificadas en el Anexo I de la consigna (ECG, PPG, EMG, acelerometría, temperatura, presión, nivel, corriente, posición angular y glucosa/SpO₂), cada una con sus propiedades espectrales características requerimiento de fase lineal, ancho de banda de transición, tipo de respuesta en frecuencia documentadas como conocimiento de partida. A esto se suma un catálogo de plataformas de microcontrolador (ESP32, STM32F4, Arduino UNO, ATtiny85, o una configuración personalizada) con sus recursos reales de memoria, cómputo y presencia de unidad de punto flotante en hardware.

Todos los campos de entrada tanto los de la señal (frecuencia de muestreo, requerimiento de fase lineal, nivel de ruido, latencia, pendiente de transición) como los del microcontrolador (RAM, Flash, MIPS) se completan automáticamente al seleccionar una opción o plataforma predefinida, pero permanecen totalmente editables por el usuario. Esto permite tanto evaluar los escenarios de referencia del enunciado como explorar combinaciones intermedias o hipotéticas que no figuran en ninguna tabla, sin perder la trazabilidad del criterio de decisión.

La decisión de arquitectura no surge de una tabla de consulta estática, sino de un motor de reglas ponderadas siguiendo el criterio de referencia del punto 4.4 del enunciado (fase lineal, límites de memoria, estabilidad numérica, exigencia de la pendiente de transición) que estima el orden real necesario del filtro mediante la aproximación de Kaiser, verifica su factibilidad contra la memoria disponible en la plataforma seleccionada, y solo entonces resuelve entre FIR e IIR. Esto permite que el agente identifique casos límite reales: por ejemplo, una plataforma con recursos ligeramente superiores a los previstos puede sostener una arquitectura de mayor orden y preservar fase lineal donde una implementación más conservadora recurriría a IIR, y el agente lo señala explícitamente junto con el trade-off detectado una funcionalidad que excede el requisito de trazabilidad del punto 31 y que se usó activamente durante la validación del punto 4.5 para verificar consistencia con el análisis de los Pasos 1 a 3.

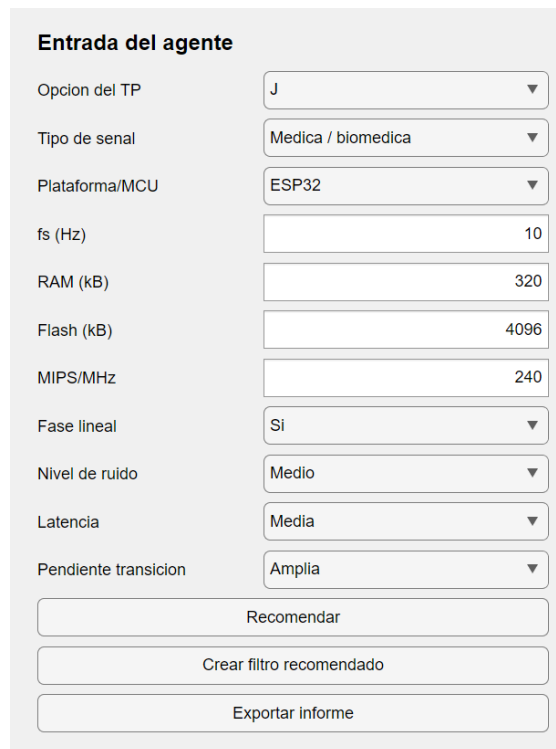
Una vez obtenida la recomendación con su justificación por etapa, el usuario no queda limitado al resultado sugerido: puede intervenir sobre el diseño y modificar manualmente el orden, la frecuencia de corte o el tipo de filtro de cada etapa según su propio criterio de ingeniería, antes de generar el filtro definitivo. El agente calcula entonces los coeficientes reales y presenta la respuesta en frecuencia resultante (magnitud, fase y retardo de grupo) con las especificaciones de diseño superpuestas para facilitar su interpretación. Todo el proceso, incluyendo entradas, reglas activadas, coeficientes por etapa y gráficas, puede

exportarse como informe completo.

Finalmente, el agente no se limita al diseño en simulación: incorpora una conexión directa por puerto serie con el microcontrolador físico, bajo un protocolo binario propio, que transfiere los coeficientes calculados al hardware embebido y abre a continuación una ventana de monitoreo en tiempo real, donde se visualizan de forma dinámica la señal cruda y la filtrada superpuestas junto con el espectro en frecuencia (FFT) de ambas cerrando el ciclo completo entre el diseño teórico, la posibilidad de ajuste manual, y su verificación empírica sobre el sistema embebido real.

En este link [<https://github.com/mpenon4/agente-filtros-digitales>] se encuentra el repositorio del folder completo donde se tiene todos los archivos del agente a explicar. También se tiene información más específica de la estructura y función de cada archivo .m dentro del folder y diagrama de flujo.

8.1 Variables de entrada del agente



El formulario, titulado "Entrada del agente", contiene los siguientes campos:

Variable	Valor
Opcion del TP	J
Tipo de señal	Medica / biomedica
Plataforma/MCU	ESP32
fs (Hz)	10
RAM (kB)	320
Flash (kB)	4096
MIPS/MHz	240
Fase lineal	Si
Nivel de ruido	Medio
Latencia	Media
Pendiente transicion	Amplia

Debajo de los campos, hay tres botones:

- Recomendar
- Crear filtro recomendado
- Exportar informe

Fig 13. Variables de entrada en el agente

Como puede observarse, resulta primordial determinar en primer lugar qué tipo de señal se necesita filtrar y sobre qué microcontrolador se va a implementar, para luego definir los valores asociados que condicionan la decisión: requerimiento de fase lineal, restricción de latencia, nivel de ruido relativo, y demás variables descriptas.

Un aspecto central del agente es que, más allá de las diez opciones predefinidas del Anexo I, cuenta con una configuración de tipo Personalizado, tanto para el tipo de señal como para la plataforma de hardware. Esto permite completar libremente cada uno de los campos de entrada y observar la recomendación que el sistema genera para esa combinación particular, sin quedar limitado a los escenarios de referencia planteados por la cátedra.

8.2 Salidas esperadas del agente

Entrada del agente

Opción del TP: J
Tipo de señal: Médica / biomédica
Plataforma/MCU: ESP32
fs (Hz): 10
RAM (kB): 320
Flash (kB): 4096
MIPS/MHz: 240
Fase lineal: SI
Nivel de ruido: Medio
Latencia: Media
Pendiente transición: Amplia

Resultado: FIR + IIR

CGM/SpO2: suavizado fisiológico y remoción de deriva/DC. | FIR: FIR directo con buffer circular | IIR: SOS / Direct Form II transposed

Confianza: 71.4 %

Etapa 1: la referencia usa FIR; se toma como sesgo base y orden de referencia si aplica.
Etapa 1: fase lineal solicitada por el usuario en una etapa compatible con FIR; se suma peso 3 a FIR.
Etapa 1: FIR estimado de orden 200 usa 1.57 kB; cadena completa 1.67 de 320.00 kB disponibles.
Etapa 2: la referencia usa IIR; se mantiene como sesgo base salvo que recursos/fase justifiquen cambiar.
Etapa 2: fase lineal global activada, pero la etapa de referencia es IIR; no se convierte automáticamente a FIR.
Etapa 2: FIR estimado de orden 12 usa 0.10 kB; cadena completa 1.67 de 320.00 kB disponibles.

1) FIR LOWPASS, orden 200, f=0.5 Hz. FIR lowpass calculado por reglas ponderadas para CGM/SpO2, orden 200, f=0.5 Hz. Método propuesto: ventana Hamming; orden máximo sugerido: 80.
2) IIR HIGHPASS, orden 1, f=0.01 Hz. IIR highpass calculado por reglas ponderadas para CGM/SpO2, orden 1, f=0.01 Hz. Método propuesto: Butterworth/notch en SOS para estabilidad numérica.

Ajustes antes de crear

Etapas	Tipo	Respuesta	Orden	Frecuencia Hz
Etapas 1/2 ...	FIR	lowpass	200	0.5

Recomendar
Crear filtro recomendado
Exportar informe

Fig 14. Respuesta y recomendación del agente para caso señal opción J

La Figura 14 muestra la salida generada por el agente para el caso analizado (Opción J: señal de glucosa/SpO₂, plataforma ESP32), donde pueden identificarse directamente los cuatro elementos requeridos por la consigna:

Recomendación (FIR o IIR): El resultado se presenta de forma inmediata en el encabezado de la sección derecha: *"Resultado: FIR + IIR"*. Dado que la Opción J corresponde a una cadena de dos etapas en cascada (suavizado FIR y remoción de deriva/DC en IIR), el agente indica explícitamente el tipo resuelto para cada una, en lugar de una única recomendación global — lo cual es coherente con que se trata de dos filtros con requisitos de diseño distintos, no de un único bloque.

Estructura sugerida: Se muestra a continuación de la recomendación: *"FIR: FIR directo con buffer circular | IIR: SOS / Direct Form II transposed"*. Esto responde puntualmente a lo pedido en el enunciado (Direct Form I/II, cascada SOS), indicando además el criterio de implementación concreto que se usará en el firmware embebido para cada etapa.

Justificación técnica / tabla de reglas activadas. Se presenta en dos niveles de detalle complementarios:

- El listado superior muestra, regla por regla, el razonamiento del motor de decisión ponderado: qué peso se activó, por qué (por ejemplo, *"fase lineal solicitada por el usuario en una etapa compatible con FIR; se suma peso 3 a FIR"*), y la verificación de factibilidad de recursos realizada (*"FIR estimado de orden 200 usa 1.57 kB; cadena completa 1.67 de 320.00 kB disponibles"*), evidenciando que la recomendación no es una consulta a tabla sino el resultado de una evaluación cuantitativa contra los recursos reales de la plataforma seleccionada.
- El cuadro inferior resume, en lenguaje natural, el criterio de diseño de cada etapa: tipo, orden, frecuencia de corte y método de diseño propuesto (ventana de Hamming para el FIR, estructura SOS Butterworth para el IIR), incluyendo el orden máximo sugerido como referencia de diseño.

Nivel de confianza: Se informa como porcentaje explícito (*"Confianza: 71.4%"*), calculado como la proporción del puntaje ponderado obtenido por la recomendación ganadora sobre el puntaje total evaluado entre FIR e IIR — no es un valor fijo ni binario, sino que varía según cuán unánimemente las reglas activadas coincidan en una misma dirección para el caso analizado.

Cabe destacar que, sobre esta recomendación, el agente habilita además un panel de ajuste manual ("*Ajustes antes de crear*"), donde el usuario puede modificar el tipo, la respuesta en frecuencia, el orden o la frecuencia de corte de cada etapa antes de generar el filtro definitivo, incorporando así el criterio de diseño propio del ingeniero sobre la base de la recomendación automática.

Respuesta en frecuencia filtro recomendado

Luego de la recomendación específica, y el criterio del ingeniero para con los órdenes, frecuencias y tipos de filtros por etapas estén definidos. Se puede apretar el botón "crear filtro recomendado".

De esta manera y en este caso específico para una señal opción J: Señal de glucosa podremos obtener unos filtros que coinciden también con lo calculado por nuestra parte en secciones VI y VII.

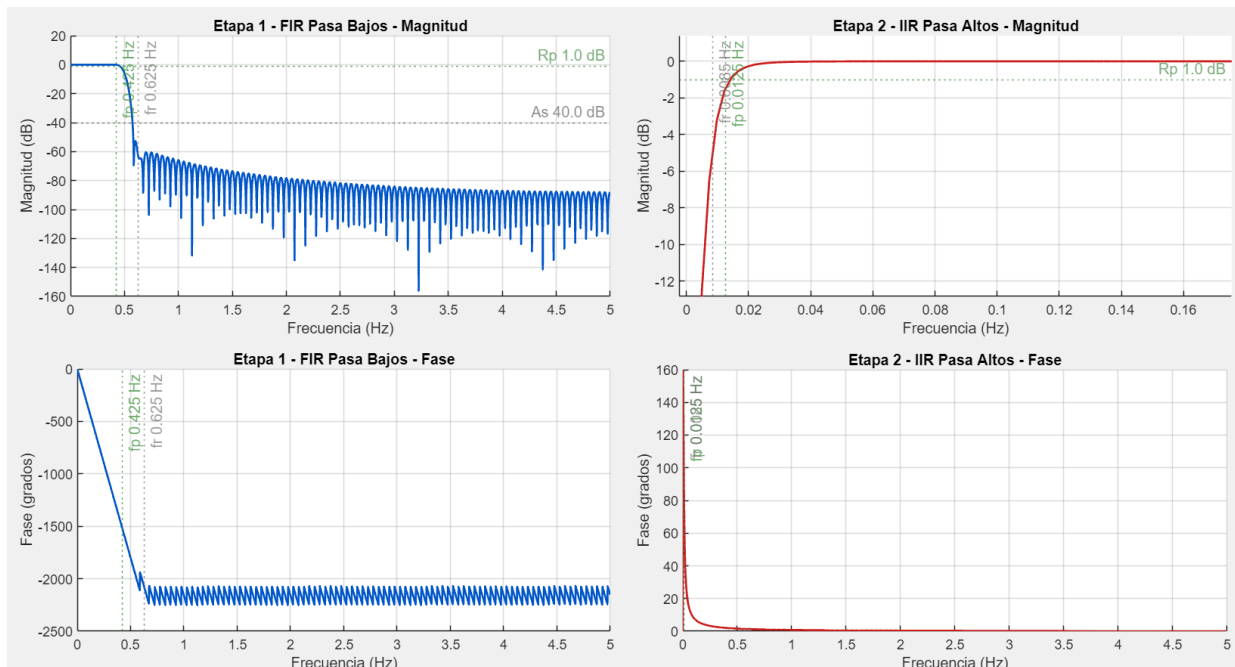


Fig 15. Respuesta en frecuencia filtros recomendados

Asimismo, y de gran relevancia para la sección posterior de implementación en software embebido, el agente incorpora la opción de **exportar informe**. Mediante esta funcionalidad se genera un archivo con toda la información relevante del filtro diseñado (variables de entrada, recomendación, reglas activadas, coeficientes por etapa y especificaciones de diseño), quedando disponible como respaldo documental del proceso y como insumo directo para la etapa de implementación en el microcontrolador. A continuación se presenta un ejemplo del contenido exportado para el caso específico analizado, disponible para su consulta en el siguiente enlace:

https://docs.google.com/document/d/1-AY2MO-fnzBGU_HC7Wt4VJ9-chbmyoPf6cIXEjeK4OY/edit?usp=sharing

8.3 Implementacion del agente

Estrategia seleccionada: sistema de reglas IF-THEN ponderadas

De las cinco estrategias propuestas por la consigna, se optó por un **sistema de reglas IF-THEN**, implementado en MATLAB dentro de la función `recomendar_filtro.m`. A diferencia de un sistema de reglas binario tradicional (donde cada condición dispara una acción fija), el motor aquí desarrollado acumula un puntaje ponderado por cada regla activada (diferenciado para la hipótesis FIR y para la hipótesis IIR) y resuelve la recomendación final por comparación de puntajes acumulados, no por la primera regla que se cumple. Esto lo distingue de un árbol de decisión (donde la trayectoria de evaluación es jerárquica y excluyente) y de la lógica difusa (donde las variables tendrían grados de pertenencia continuos en lugar de condiciones booleanas discretas como las aquí utilizadas).

Se descartó el uso de una red neuronal por la complejidad y generarlos sintéticamente a partir de las mismas reglas de diseño que se buscaba automatizar habría resultado circular, sin aportar generalización real. Se descartó asimismo la interfaz con una IA generativa externa como motor de decisión, dado que introduce opacidad y no determinismo en un dominio (selección FIR/IIR) que tiene fundamentos analíticos bien establecidos en la teoría de procesamiento digital de señales, y que la consigna exige documentar con trazabilidad explícita (punto 31).

Ventajas de la estrategia para este caso de uso

La principal ventaja es la **trazabilidad completa**: cada recomendación puede reconstruirse regla por regla, lo cual resulta indispensable en un dominio médico-embebido donde una decisión de arquitectura de filtro debe poder auditarse y justificarse técnicamente, no solo predecirse. Adicionalmente, el sistema no se limita a evaluar reglas categóricas: incorpora una **estimación cuantitativa real** del orden de filtro necesario mediante la aproximación de Kaiser, y verifica su factibilidad contra la memoria disponible en la plataforma de destino antes de decidir evitando que la recomendación ignore restricciones de recursos reales del hardware embebido.

Limitaciones frente a las alternativas

Frente a un árbol de decisión entrenado, el sistema de reglas requiere que los pesos y umbrales sean definidos manualmente por el diseñador, sin posibilidad de ajuste automático a partir de datos. Frente a la lógica difusa, pierde granularidad en los límites entre categorías (por ejemplo, la transición entre "pendiente estrecha" y "amplia" es discreta, no continua), lo cual puede generar decisiones abruptas en casos límite cercanos a un umbral. Frente a una red neuronal, no generaliza a combinaciones de variables no contempladas explícitamente en las reglas.

Mecanismo de confianza

La confianza se calcula como la proporción del puntaje ponderado obtenido por la clase ganadora sobre el puntaje total evaluado: $\text{confianza} = 100 \times \max(p_{\text{FIR}}, p_{\text{IIR}}) / (p_{\text{FIR}} + p_{\text{IIR}})$. Se acota entre 55% y 97% en lugar de 0-100% porque los extremos de ese rango no son informativos en este dominio: un valor cercano a 100% sugeriría una certeza absoluta que el sistema no puede garantizar dado que las reglas son heurísticas de ingeniería, no una prueba formal; y un valor cercano al 50% (empate real) ya se resuelve mediante el criterio de desempate por factibilidad de RAM, por lo que nunca debería

reportarse como resultado final indefinido.

Justificación de la aproximación de Kaiser

Se utilizó la fórmula de Kaiser para estimar el orden del filtro FIR por ser una aproximación estándar, simple y ampliamente documentada en la bibliografía de procesamiento digital de señales, que permite obtener una cota de orden razonable a partir de únicamente tres parámetros de diseño (atenuación de banda de rechazo, ancho de banda de transición y frecuencia de muestreo), sin requerir iteración numérica. Si bien existen métodos más precisos (como el algoritmo de Parks-McClellan/Remez), estos exigen resolver un problema de optimización que no aporta valor adicional en esta etapa de pre-selección arquitectónica el orden final exacto se recalcula de todas formas con **designfilt** una vez que el usuario confirma o ajusta la etapa a diseñar.

Diagrama de flujo general

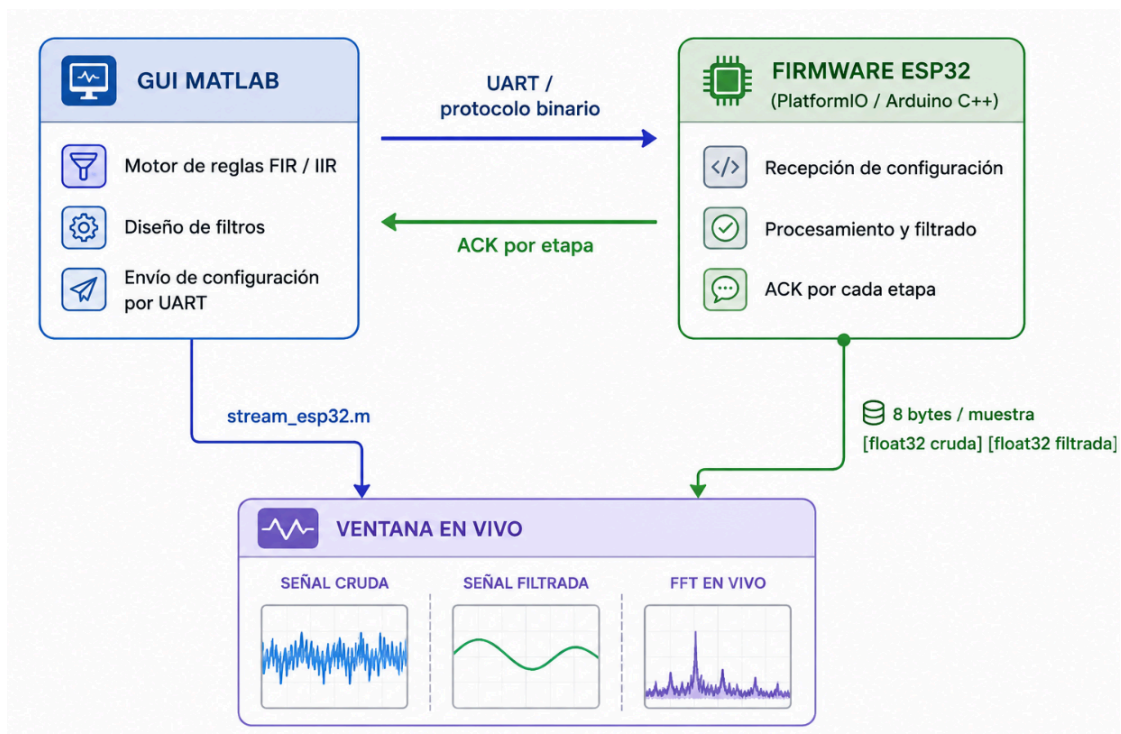


Fig 16. Diagrama de flujo de datos

Validación del agente

Con el objetivo de validar el funcionamiento del agente de diseño y del sistema de filtrado desarrollado, se definieron distintos casos de prueba correspondientes a aplicaciones reales de procesamiento digital de señales. En cada caso, el agente recibe la descripción del escenario de adquisición y genera automáticamente la configuración necesaria para el diseño del filtro. Posteriormente, la GUI implementa el filtro en MATLAB, verifica su respuesta en frecuencia y finalmente envía los coeficientes al microcontrolador seleccionado para su ejecución en tiempo real.

A continuación se presenta el primer caso de estudio.

Caso de prueba 1 – Señal ECG: eliminación de interferencia de red y ruido muscular

Para este caso se considera una señal de electrocardiograma (ECG), cuya información útil se encuentra principalmente entre 0,5 Hz y 40 Hz. El objetivo es eliminar la interferencia de la red eléctrica (50 Hz) y el ruido muscular de alta frecuencia, preservando la morfología de las ondas características del ECG.

Debido a los requerimientos computacionales del filtrado, se seleccionó como plataforma de implementación un **STM32**, cuyas características de procesamiento permiten ejecutar el algoritmo con amplio margen temporal.

Fig 17. Resultado del ejemplo seleccionando un STM32

Para demostrar que realmente tiene en cuenta los valores de microcontroladores y demás, vemos que cambiando de opción a un Arduino UNO se reconfigura de la siguiente manera:

Fig 18. Resultado usando un Arduino UNO como micro

Volviendo al caso principal dado con un STM32, le damos a crear filtro para ver si es correcto:

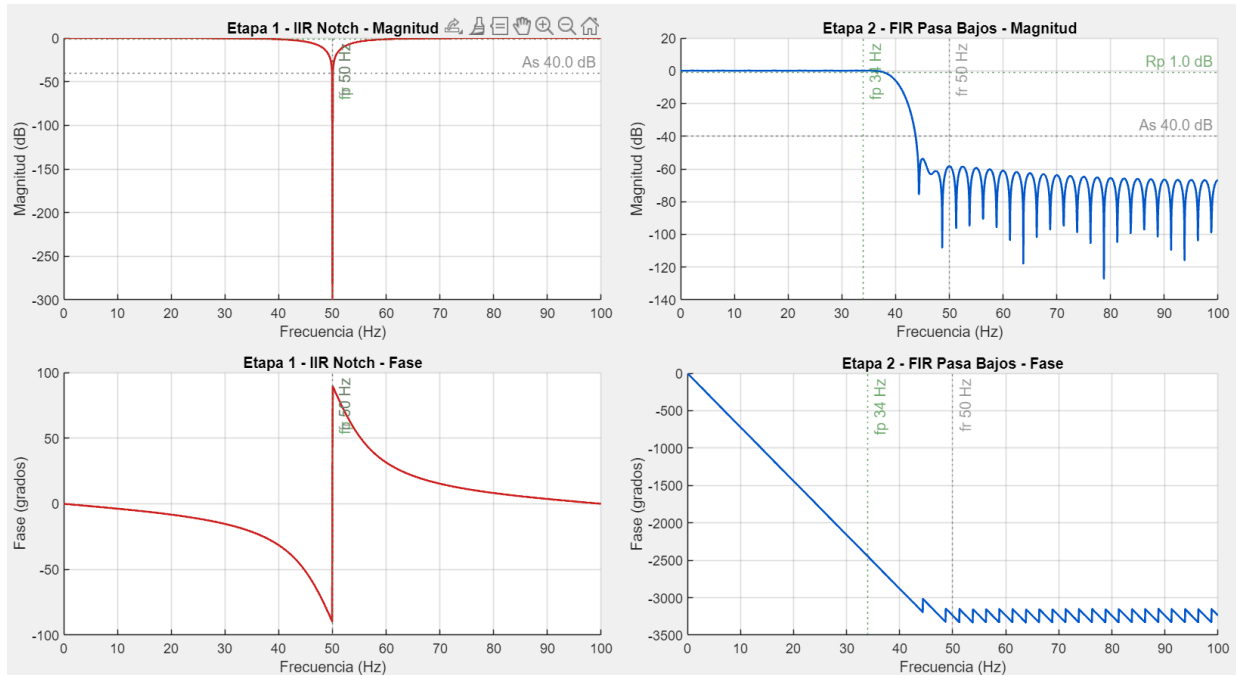


Fig 18. Etapas de filtros creados para señal tipo A con STM32

Caso de prueba 2 – Monitoreo de condición de rodamientos mediante acelerómetro MEMS

En este segundo caso de prueba se considera un sistema de monitoreo de condición basado en un acelerómetro MEMS instalado sobre un motor o rodamiento. La señal adquirida presenta una componente continua asociada a la gravedad, vibraciones mecánicas de interés entre **10 Hz y 200 Hz**, interferencias electromagnéticas de alta frecuencia y ruido introducido por el proceso de digitalización.

Para el procesamiento de esta señal se seleccionó un **ESP32**, debido a su mayor capacidad de procesamiento y memoria, permitiendo implementar filtros FIR e IIR con amplio margen temporal para una frecuencia de muestreo de **1 kHz**.

A partir de este escenario, el agente genera automáticamente la configuración necesaria para el diseño de un filtro **pasa banda FIR (10–200 Hz)** destinado a aislar la banda de vibración de interés, junto con un **filtro pasa bajos IIR** de preacondicionamiento para atenuar las componentes de alta frecuencia. Posteriormente, la GUI implementa ambos filtros en MATLAB, analiza su respuesta en frecuencia y prepara los coeficientes para su envío al microcontrolador.

Entrada del agente

Opcion del TP: B

Tipo de señal: Vibracion mecanica

Plataforma/MCU: ESP32

fs (Hz): 1000

RAM (kB): 320

Flash (kB): 4096

MIPS/MHz: 240

Fase lineal: No

Nivel de ruido: Alto

Latencia: Media

Pendiente transicion: Estrecha

Recomendar

Crear filtro recomendado

Exportar informe

Resultado: FIR + IIR

Acelerometro MEMS: banda de vibracion y acondicionamiento anti-aliasing. | FIR: FIR directo con buffer circular | IIR: SOS / Direct Form II transposed

Confianza: 60.0 %

Etapa 1: la referencia usa FIR: se toma como sesgo base y orden de referencia si aplica.

Etapa 1: fase lineal no solicitada por el usuario; no se suma peso a FIR por fase.

Etapa 1: transicion estrecha con computo disponible; se suma peso 1 a IIR por menor orden.

Etapa 1: FIR estimado de orden 60 usa 0.48 kB; cadena completa 0.58 de 320.00 kB disponibles.

Etapa 1: ruido alto; mantener A_s de referencia en 40 dB y validar SNR.

Etapa 2: la referencia usa IIR: se mantiene como sesgo base salvo que recursos/fase justifiquen cambiar.

1) FIR BANDPASS, orden 60, f=[0.5 20] Hz. FIR bandpass calculado por reglas ponderadas para acelerometro MEMS, orden 60, f=[0.5 20] Hz. Metodo propuesto: ventana Hamming; orden maximo sugerido: 80.

2) IIR LOWPASS, orden 2, f=20 Hz. IIR lowpass calculado por reglas ponderadas para acelerometro MEMS, orden 2, f=20 Hz. Metodo propuesto: Butterworth/hoitch en SOS para estabilidad numerica.

Ajustes antes de crear

Etapa	Tipo	Respuesta	Orden	Frecuencia Hz
Etapa 1/2 ~...	FIR	bandpass	60	[0.5 20]

Fig 19. Configuración de filtro para señal tipo B con ESP32

Y visualizamos el filtro creado para este caso:

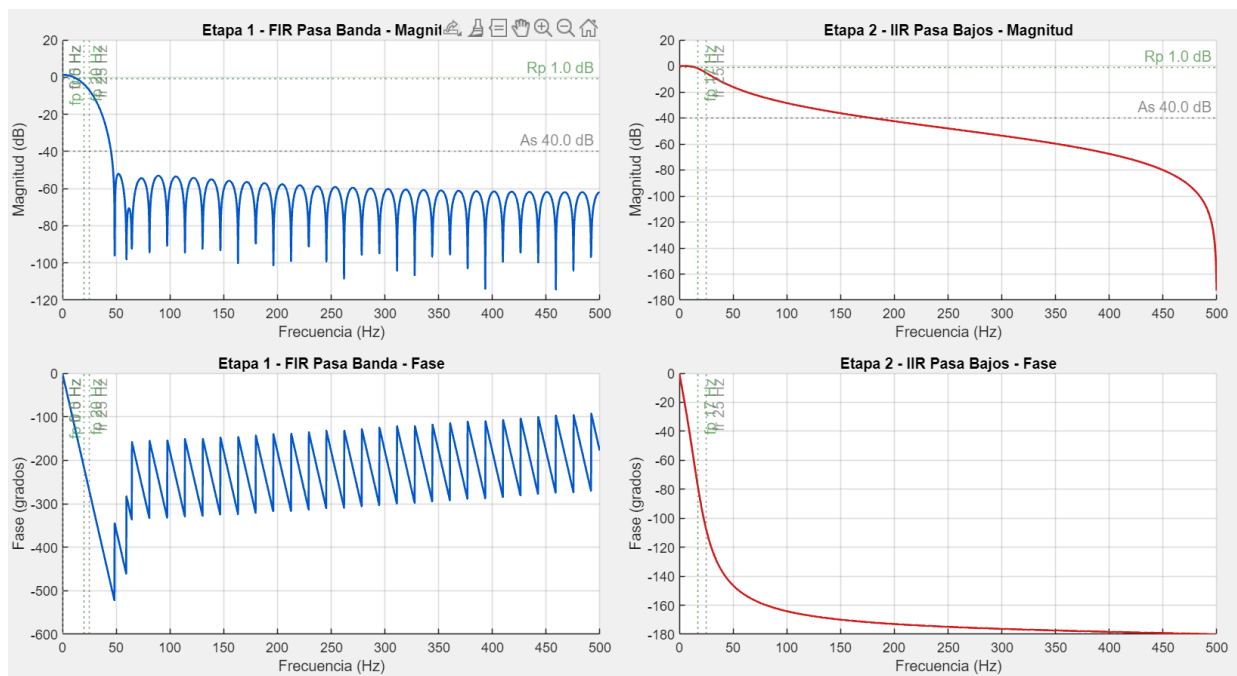


Fig 20. Filtro creado para señal tipo B con ESP32

Caso de prueba 3 – Detección de activación muscular mediante señales EMG

En este tercer caso de prueba se analiza una señal de **electromiografía de superficie (EMG)**, utilizada para detectar la activación de los músculos y empleada en aplicaciones como el control de prótesis, exoesqueletos e interfaces hombre-máquina. La señal adquirida contiene componentes útiles entre **20 Hz y 500 Hz**, además de interferencia de la red eléctrica, artefactos de movimiento de baja frecuencia y ruido eléctrico.

Para este escenario se seleccionó un **ESP32**, ya que su capacidad de procesamiento permite implementar el procesamiento en cascada requerido para este tipo de señales con una frecuencia de muestreo de **1 kHz**.

A partir de este caso de estudio, el agente genera automáticamente la configuración necesaria para implementar la cadena de procesamiento compuesta por un **filtro pasa banda**, una **etapa de rectificación de onda completa** y un **filtro pasa bajos** destinado a obtener la envolvente de activación muscular. Posteriormente, la GUI diseña los filtros en MATLAB, verifica su respuesta en frecuencia y prepara los coeficientes para su implementación en el microcontrolador.

Entrada del agente

Opcion del TP: D

Tipo de señal: Medica / biomedica

Plataforma/MCU: ESP32

fs (Hz): 1000

RAM (kB): 320

Flash (kB): 4096

MIPS/MHz: 240

Fase lineal: No

Nivel de ruido: Alto

Latencia: Media

Pendiente transicion: Estrecha

Recomendar

Crear filtro recomendado

Exportar informe

Envío opcional al ESP32

Resultado: FIR + IIR

EMG superficial: banda muscular y envolvente lenta. | FIR: FIR directo con buffer circular | IIR: SOS / Direct Form II transposed

Confianza: 60.0 %

Etapa 1: la referencia usa FIR, se toma como sesgo base y orden de referencia si aplica.
Etapa 1: fase lineal no solicitada por el usuario; no se suma peso a FIR por fase.
Etapa 1: transición estrecha con computo disponible; se suma peso 1 a IIR por menor orden.
Etapa 1: FIR estimado de orden 60 usa 0.48 kB; cadena completa 0.58 de 320.00 kB disponibles.
Etapa 1: ruido alto; mantener A_s de referencia en 40 dB y validar SNR.
Etapa 2: la referencia usa IIR; se mantiene como sesgo base salvo que recursos/fase justifiquen cambiar.

1) FIR BANDPASS, orden 60, f=[20 450] Hz. FIR bandpass calculado por reglas ponderadas para EMG, orden 60, f_c=[20 450] Hz. Metodo propuesto: ventana Hamming; orden maximo sugerido: 80.
2) IIR LOWPASS, orden 2, f=5 Hz. IIR lowpass calculado por reglas ponderadas para EMG, orden 2, f_c=5 Hz. Metodo propuesto: Butterworth/otch en SOS para estabilidad numerica.

Ajustes antes de crear

Etapa	1/2 ...	Tipo	Respuesta	Orden	Frecuencia Hz
Etapa 1	2 ...	FIR	bandpass	60	[20 450]

Coefficientes exportables

Fig 21. Resultado de configuración de filtro para señal tipo D con ESP32

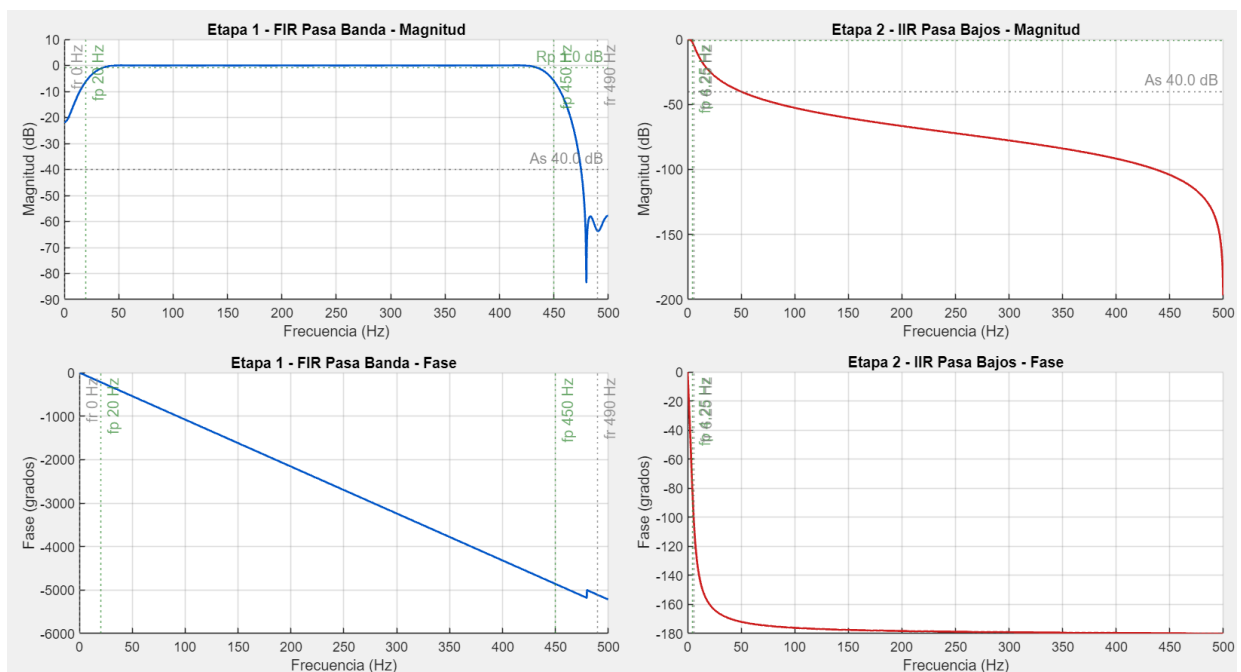


Fig 22. Filtro creado para señal tipo D con agente

IX_ Implementación Embebida

9.1 Descripción del hardware

La plataforma utilizada para la implementación fue un ESP32-WROOM-32, por su capacidad de cómputo, soporte de UART, ADC integrado y disponibilidad de aritmética en punto flotante simple por hardware. El módulo integra un microcontrolador Xtensa LX6 dual-core de 32 bits, con frecuencia de reloj de hasta 240 MHz, alrededor de 520 kB de SRAM interna y memoria Flash externa del módulo; además, dispone de ADC de 12 bits y múltiples canales analógicos.

En este trabajo se priorizó la validación embebida sobre una plataforma con FPU, por lo que los coeficientes del filtro se representan en float32, evitando la complejidad del punto fijo. Esto simplifica la implementación, reduce errores de escalado y permite una traducción directa desde los coeficientes obtenidos en MATLAB/Octave al firmware del ESP32.

El circuito de adquisición se planteó en dos modos. En la etapa de validación inicial se utiliza una señal sintética generada por software dentro del firmware, compuesta por una senoide de prueba y otras componentes de baja y alta frecuencia. En la etapa final, el mismo firmware puede adaptarse para leer una señal real por ADC desde un sensor de glucosa o desde una señal de referencia almacenada en repositorios de datos biomédicos.

9.2 Arquitectura del sistema embebido

El sistema completo se organiza como una cadena de procesamiento: adquisición de la muestra, almacenamiento temporal, filtrado digital y envío de resultados. La arquitectura general es: señal de prueba o señal real ,adquisición ,filtro digital en ESP32 ,salida por UART y por último visualización en MATLAB.

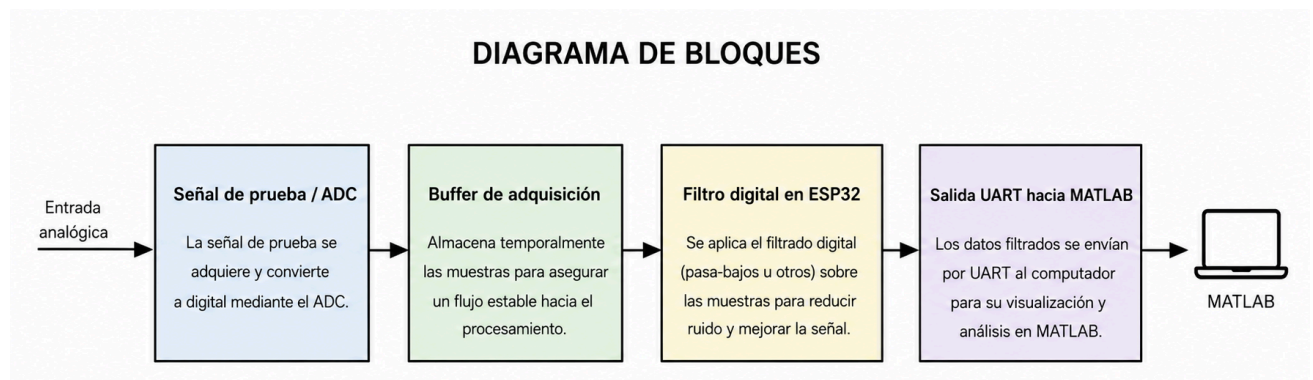


Fig 23. Diagrama de bloques del sistema completo

La estrategia de muestreo adoptada es polling no bloqueante con temporización basada en micros(). Esta decisión se justifica porque la frecuencia de muestreo usada en la validación es baja y el control por software es suficiente para mantener una tasa estable sin introducir la complejidad de una interrupción por timer o DMA.

El manejo de datos usa un buffer circular para la etapa FIR, ya que permite insertar nuevas muestras sin mover el resto del arreglo y conservar la historia necesaria para la convolución. En la etapa IIR se conservan los estados internos mínimos requeridos por la ecuación en diferencias, reduciendo uso de memoria y costo computacional.

9.3 Implementación del algoritmo de filtrado

El filtro se implementó en C/C++ , con coeficientes cargados directamente en el código fuente. En lugar de depender de transferencia dinámica por serial, el agente remoto entrega el informe técnico y los coeficientes en formato textual, y esos coeficientes se copian manualmente al firmware final. Esta decisión simplifica la depuración y evita problemas de comunicación serial durante la validación.

Los coeficientes se representan en punto flotante simple (float32), porque el ESP32 dispone de FPU y no requiere aritmética de punto fijo para este caso. Por eso no es necesario realizar análisis de escalado Q-format ni corrección de overflow/underflow, salvo que luego se migre a una plataforma sin FPU como un Arduino UNO.

El algoritmo principal de filtrado puede describirse así:

1. Inicializar buffers, estados y temporizador.
2. Esperar el instante de muestreo.
3. Leer una muestra de entrada.
4. Aplicar la etapa FIR usando buffer circular.
5. Aplicar la etapa IIR sobre la salida del FIR.
6. Enviar por UART la muestra cruda y la muestra filtrada.
7. Repetir el proceso.

9.4 Validación embebida

La validación se plantea en tres niveles. Primero se usa una senoide simple de frecuencia conocida para verificar si el filtro atenúa correctamente las componentes fuera de banda y mantiene la señal útil. Después se usa una senoide muy lenta para comprobar que la banda de interés biomédica no se distorsiona. Finalmente, se propone una señal de glucosa más realista obtenida de repositorios públicos o de datos de referencia, para observar el desempeño del algoritmo en una condición cercana al uso real, esto se muestra en la sección 9.6.

Esta secuencia es mejor que una sola prueba porque demuestra el comportamiento del filtro en distintos escenarios: una prueba funcional básica, una prueba de preservación de componentes lentas y una prueba con señal fisiológica más realista

9.5 Fuentes de error

Las principales fuentes de error en la implementación son la cuantización del ADC, el truncamiento de coeficientes al pasar de doble precisión en MATLAB a float32 en el ESP32, y posibles efectos de aliasing si la frecuencia de muestreo se reduce demasiado. En esta implementación, el uso de float32 y una frecuencia de muestreo estable ayudan a minimizar esos problemas.

Otra posible fuente de error es el retardo introducido por el filtrado digital, especialmente si el FIR tiene un orden relativamente alto. Ese retardo no es un fallo del sistema, sino una consecuencia de la respuesta temporal del filtro .

La implementación embebida propuesta permite validar el filtro digital de forma progresiva: primero con una senoide simple, luego con una senoide muy lenta y finalmente con una señal de glucosa más realista. El ESP32-WROOM-32 es una plataforma adecuada para esta prueba porque dispone de ADC de 12 bits, buena capacidad de cálculo y soporte para float32 por hardware.

El enfoque de coeficientes embebidos manualmente en el firmware es coherente con el objetivo del trabajo, porque simplifica la integración, evita fallas de comunicación y deja la validación centrada en el comportamiento real del filtro.

9.6 Filtrado de señal PPG

Implementación y fuente de datos

No fue posible adquirir la señal directamente desde un sensor PPG conectado al ESP32 emisor. Por ello se utilizó un registro fisiológico real (100001_PPG) en formato WFDB compuesto por los archivos:

100001_PPG.dat: archivo binario con las muestras crudas.

100001_PPG.he: archivo de cabecera con metadatos (frecuencia de muestreo, número de muestras, ganancia, unidades).

Extracción de la señal

Se procesó el archivo .hea línea por línea mediante una expresión regular para extraer los valores numéricos necesarios. Con ello se reconstruyó un arreglo de 300 muestras reales, equivalente a 10 s de señal PPG muestreada a 30 Hz, expresadas en unidades arbitrarias (a.u.), tal como fueron registradas originalmente.

Justificación y propósito

Dado que no se dispone del sensor para la obtención de la señal ppg conectado al transmisor ESP32, fue necesario disponer de una fuente de datos que preservara la morfología fisiológica real. Por ello se optó por la reproducción cíclica del registro preexistente.

Resultado operativo

El sistema reproduce continuamente el registro real, simulando el comportamiento de un sensor PPG en tiempo real. Esto facilita la depuración y la verificación funcional del reenvío de datos hacia MATLAB y de los algoritmos de filtrado digital implementados en el receptor.

Señal PPG: es una línea casi plana alrededor de -128.9, con ondulaciones muy chicas (± 0.2 aprox.). Eso ya de por sí es esperable en una PPG: la señal tiene un componente DC grande (la absorción de luz de base por tejido, sangre venosa, hueso, etc.) y encima un componente AC muchísimo más chico (típicamente 1–2% del DC) que sube y baja con cada latido. En este caso el DC ronda -128.9 y la variación total es de apenas 0.43 unidades, consistente con esa relación AC/DC.

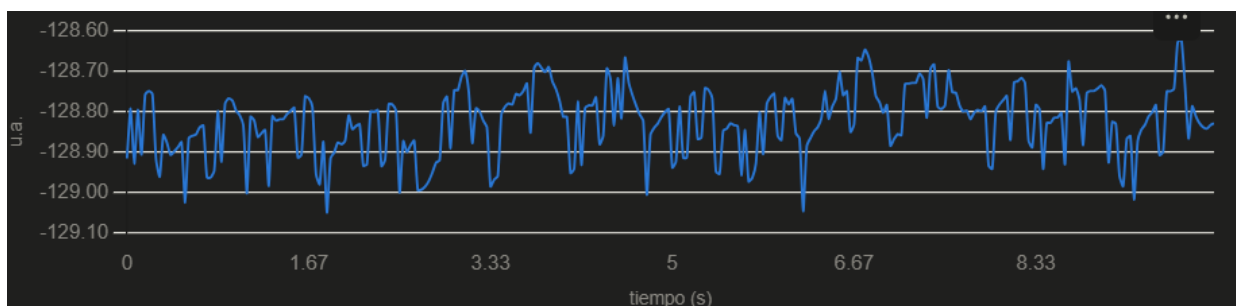


Fig 24. Señal PPG extraída de base de datos physionet.org

Señal Filtrada simulada: el pasabajos FIR (ganancia DC ≈ 1 , atenúa ruido de alta frecuencia) seguido por el pasaaltos IIR (elimina la línea de base) dejan visible únicamente el componente pulsátil, centrado en 0 y oscilando aproximadamente entre -0.11 y +0.13. Esa forma ondulada semi-periódica corresponde al pulso cardíaco aislado, lo esperado en una PPG filtrada.

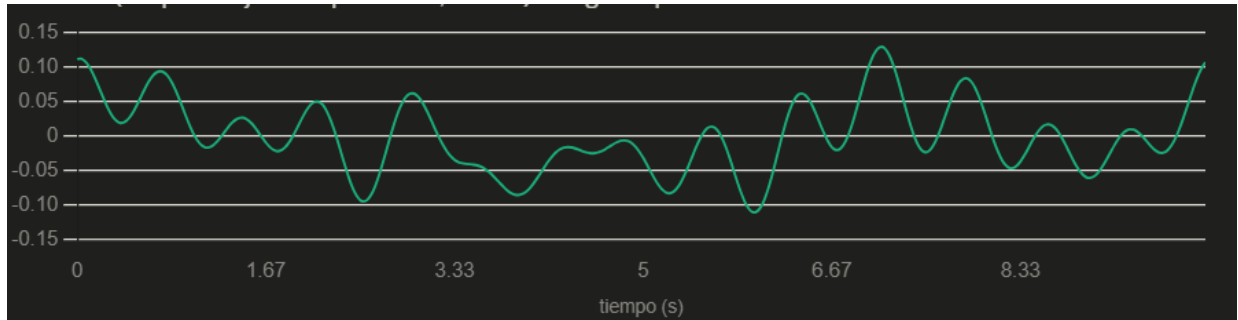


Fig 25. Simulación de señal PPG filtrada

Finalmente se obtuvieron las gráficas en MATLAB correspondientes a la señal PPG cruda, la señal después del filtrado y sus transformadas de Fourier (FFT). Las formas temporales muestran que el filtrado elimina la componente de línea base y el ruido de alta frecuencia, aislando el componente pulsátil. En el dominio de frecuencias, las FFT confirman la supresión de ruido fuera de la banda útil y la conservación de la energía en la región correspondiente al pulso cardíaco. Estos resultados coinciden estrechamente con las simulaciones previas y con lo mostrado en las figuras 24 y 25, lo que valida la implementación de filtrado y la conservación de la morfología fisiológica de la señal.

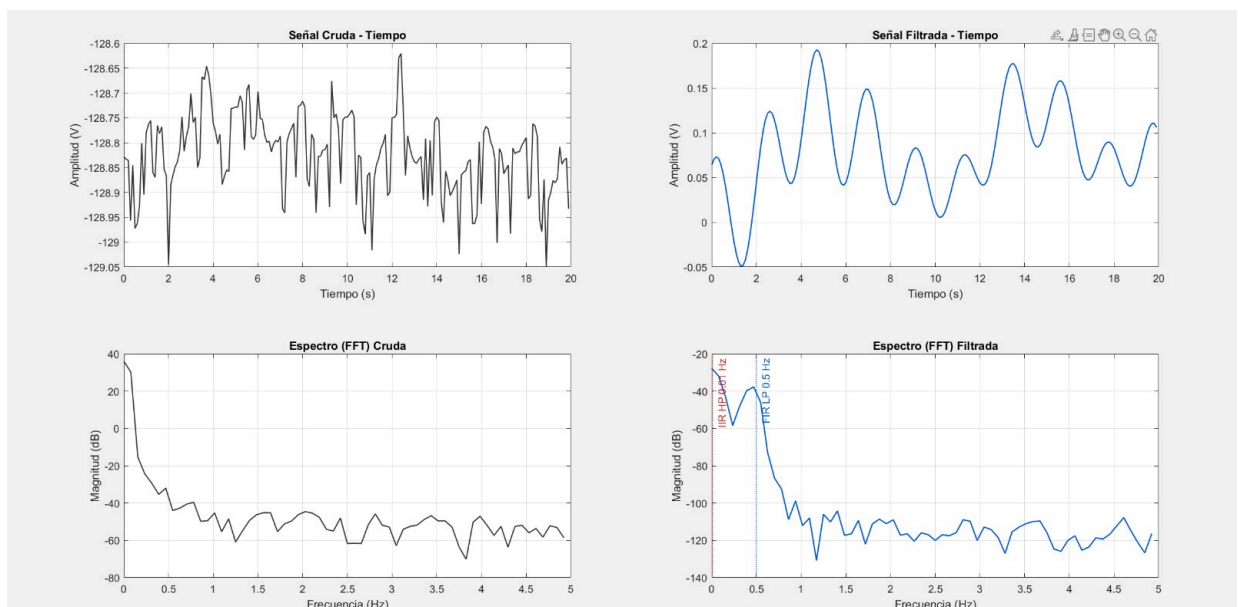


Fig 26. Señal PPG cruda (canal 1) y señal PPG filtrada (canal 2) obtenidas en MATLAB y espectros de magnitud (FFT) de la señal PPG cruda y filtrada.

X_ Conclusiones

El trabajo abordó el diseño, simulación e implementación de un sistema de filtrado digital para señales fisiológicas de baja frecuencia, desde el planteo teórico hasta las pruebas en hardware embebido.

En el diseño se verificó el comportamiento entre filtros FIR e IIR. El pasabajos FIR con ventana de Kaiser aseguró fase casi lineal y estabilidad a costa de un orden mayor ($M = 24$) y más operaciones, mientras que el pasaaltos IIR Butterworth ($N = 2$, Transformada Bilineal) eliminó la deriva con baja complejidad, introduciendo una distorsión de fase localizada. La configuración en cascada FIR + IIR resultó más conveniente que un único pasa banda, ya que permitió asignar la preservación de fase al FIR y la supresión de deriva al IIR. Las simulaciones en MATLAB cumplieron las especificaciones y mostraron una mejora de SNR de 15,07 dB en la eliminación de deriva.

El agente de decisión basado en reglas IF-THEN ponderadas permitió automatizar la selección de arquitectura manteniendo un criterio claro y revisable. La estimación de orden (método de Kaiser) y la verificación contra los recursos de cada plataforma añadieron una capa objetiva por encima de una simple tabla fija. Las pruebas con ECG, acelerometría MEMS y EMG mostraron respuestas coherentes frente a cambios de señal y de hardware.

En la implementación, el uso del ESP32-WROOM-32 y coeficientes en punto flotante facilitó el paso de MATLAB al firmware, evitando el ajuste fino de punto fijo. La validación escalonada, desde señales sintéticas hasta un registro PPG real de PhysioNet, mostró que el filtrado conserva la forma de la componente útil y reduce ruido de alta frecuencia y deriva de línea base, tanto en tiempo como en frecuencia.

Como principales limitaciones, no se contó con un sensor real de glucosa y la validación se hizo con señales sintéticas y PPG pública. Además, los coeficientes se cargaron de forma manual en el firmware, dejando pendiente el uso sistemático del canal serie desarrollado. Como trabajo futuro se plantea validar con un sensor de glucosa real, automatizar la transferencia de coeficientes vía UART e implementar el agente de decisión en plataformas más restringidas (Arduino UNO, ATtiny85) para comprobar en la práctica los casos límite.

XI_ Bibliografía

Anthropic. (2026). *Claude* (Sonnet 5) [Modelo de lenguaje de gran escala].
<https://www.anthropic.com/claude>

Google. (2026). *Antigravity* [Plataforma de desarrollo asistido por agentes de IA].
<https://antigravity.google>

MathWorks. (2024). *Digital Filter Design* (DSP System Toolbox).
<https://www.mathworks.com/help/dsp/digital-filter-design.html>

MathWorks. (2024). *designfilt*. En *MATLAB Documentation*.
<https://www.mathworks.com/help/signal/ref/designfilt.html>

MathWorks. (2024). *kaiserord*. En *MATLAB Documentation*.
<https://www.mathworks.com/help/signal/ref/kaiserord.html>

Proakis, J. G., & Manolakis, D. G. (2007). *Digital signal processing: Principles, algorithms, and applications* (4.^a ed.). Pearson Prentice Hall.

PhysioNet. (2022). *BUT PPG Database* (versión 2.0.0) [Conjunto de datos].
<https://physionet.org/content/butppg/2.0.0/>

Espressif Systems. (2023). *ESP32 series datasheet* (versión 4.6).
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf