**Sistemas y Señales II**

GUÍA N°01 GNU Radio



**Año**: 2025

**UNIVERSIDAD NACIONAL DE RÍO CUARTO**

**Facultad de Ingeniería**

**Carrera**: Ingeniería en Telecomunicaciones

**Docentes de Cátedra:**

- Ricardo Lima

- Rodrigo Prat

- Carlos Massei

- Guido Arístides Alvarez

**Alumnos:**

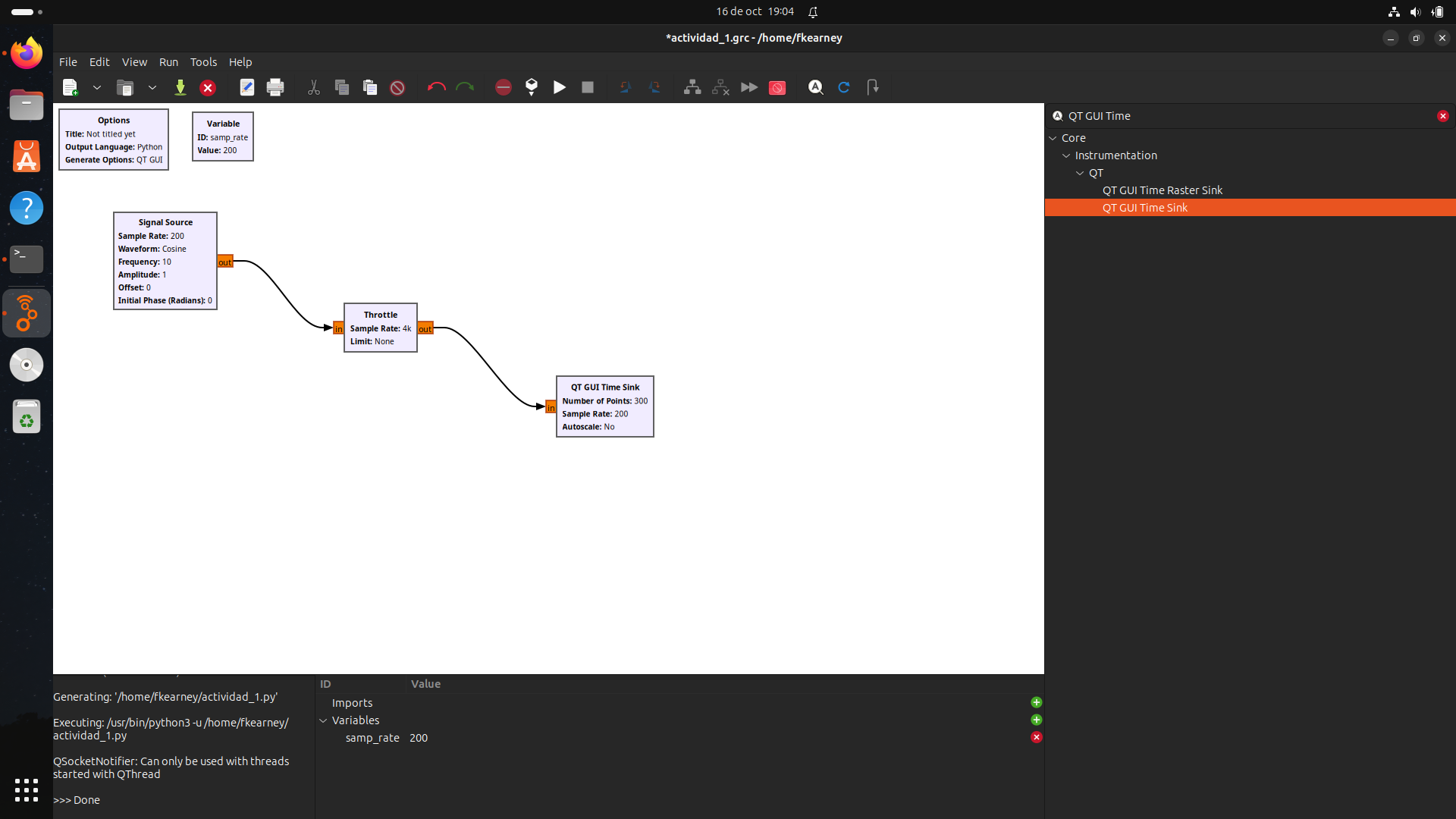
- Escobar Lautaro 45.484.501

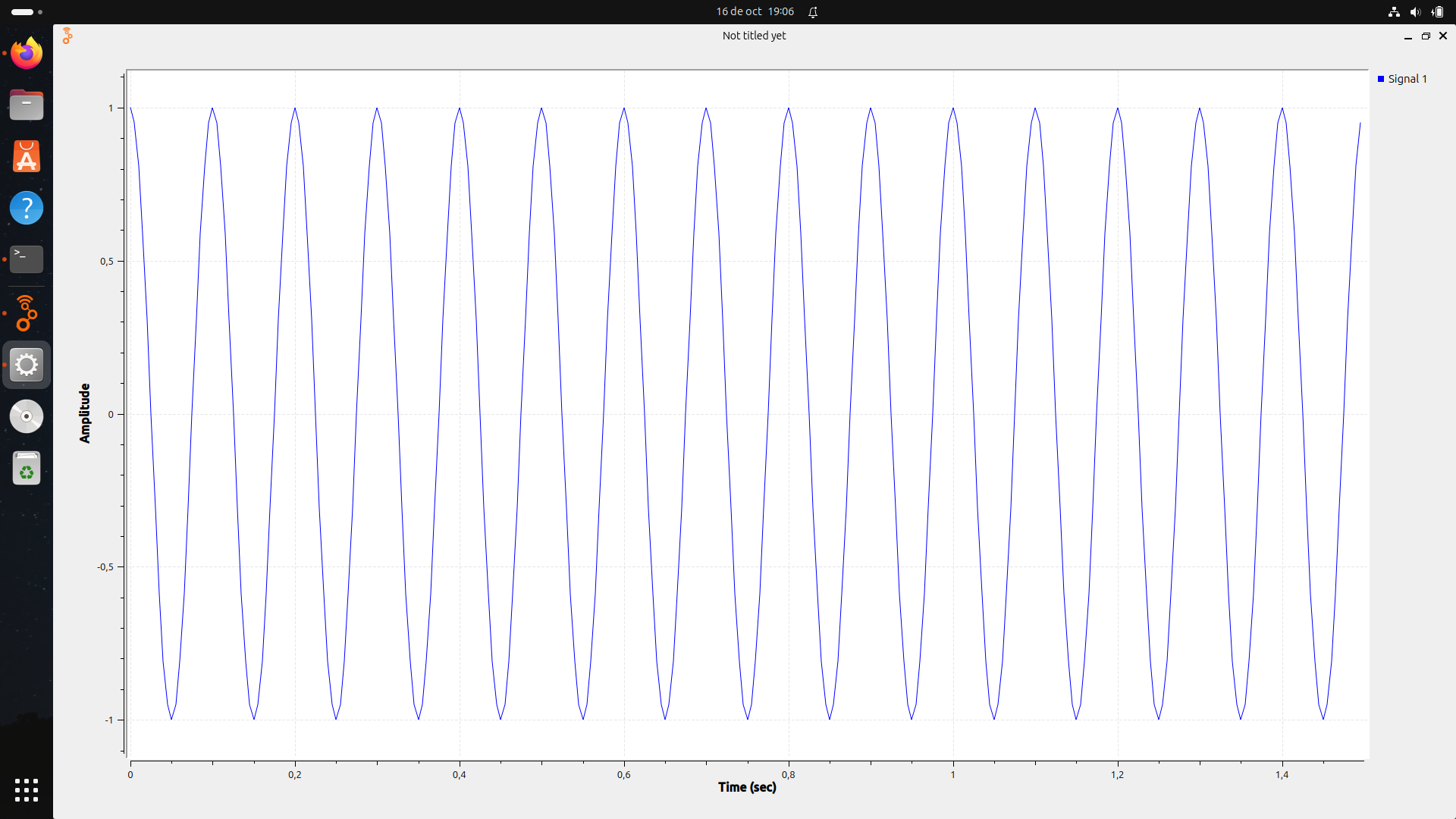
- Kearney Francisco 45.593.688

**Ejercicio 1:**

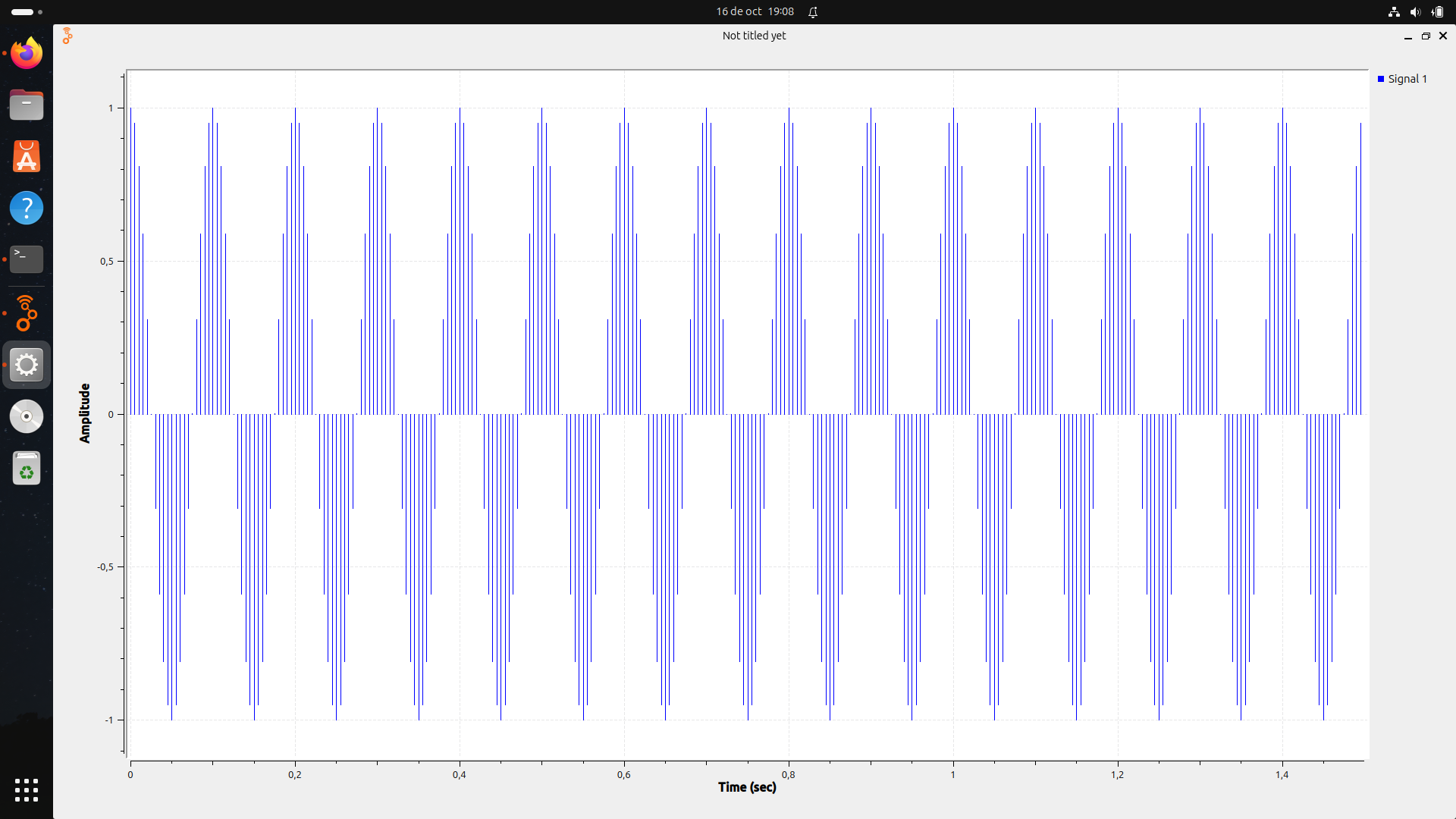
2. Implementación de las señales y en GNU Radio:

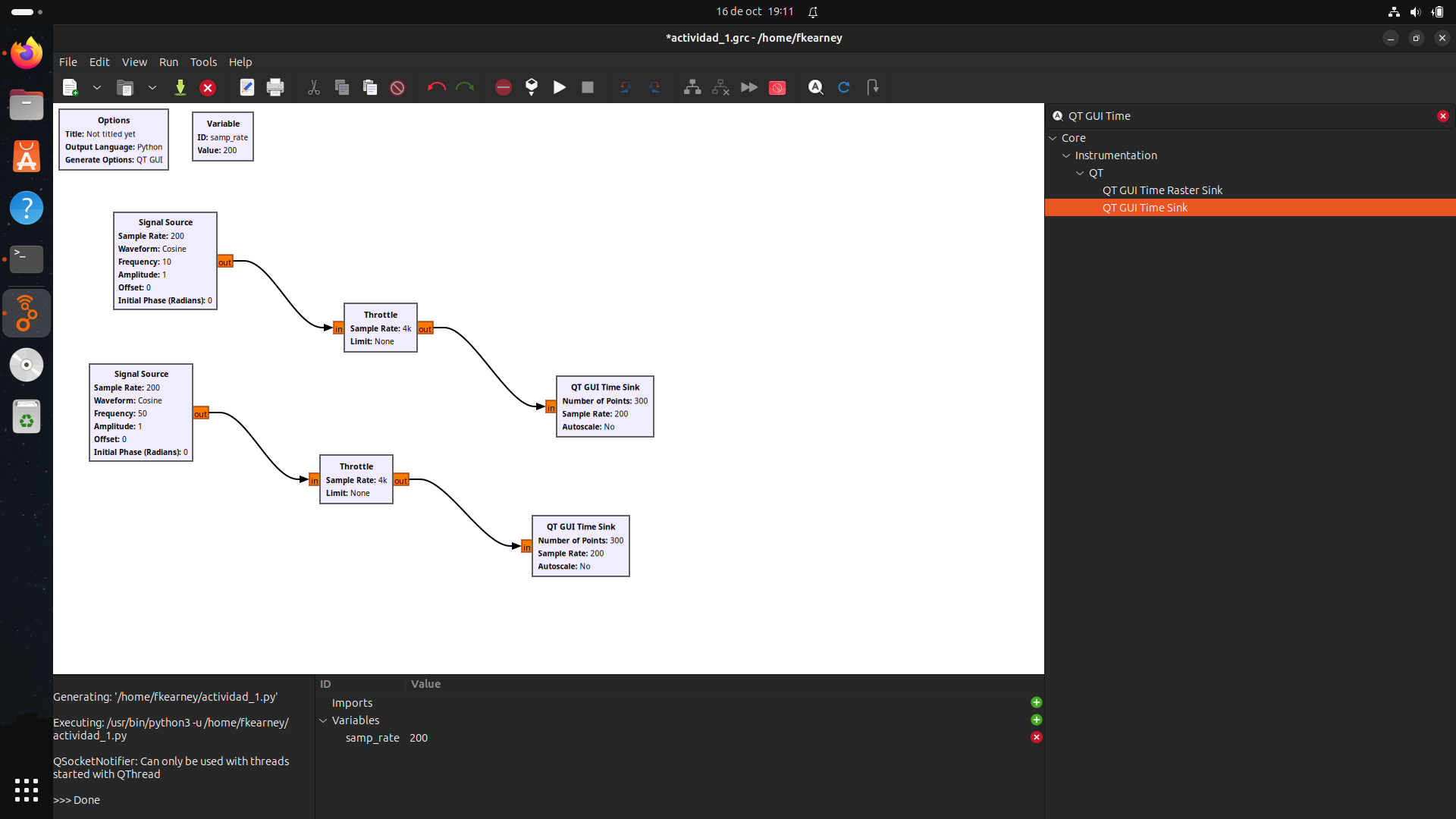
-Señal 1():

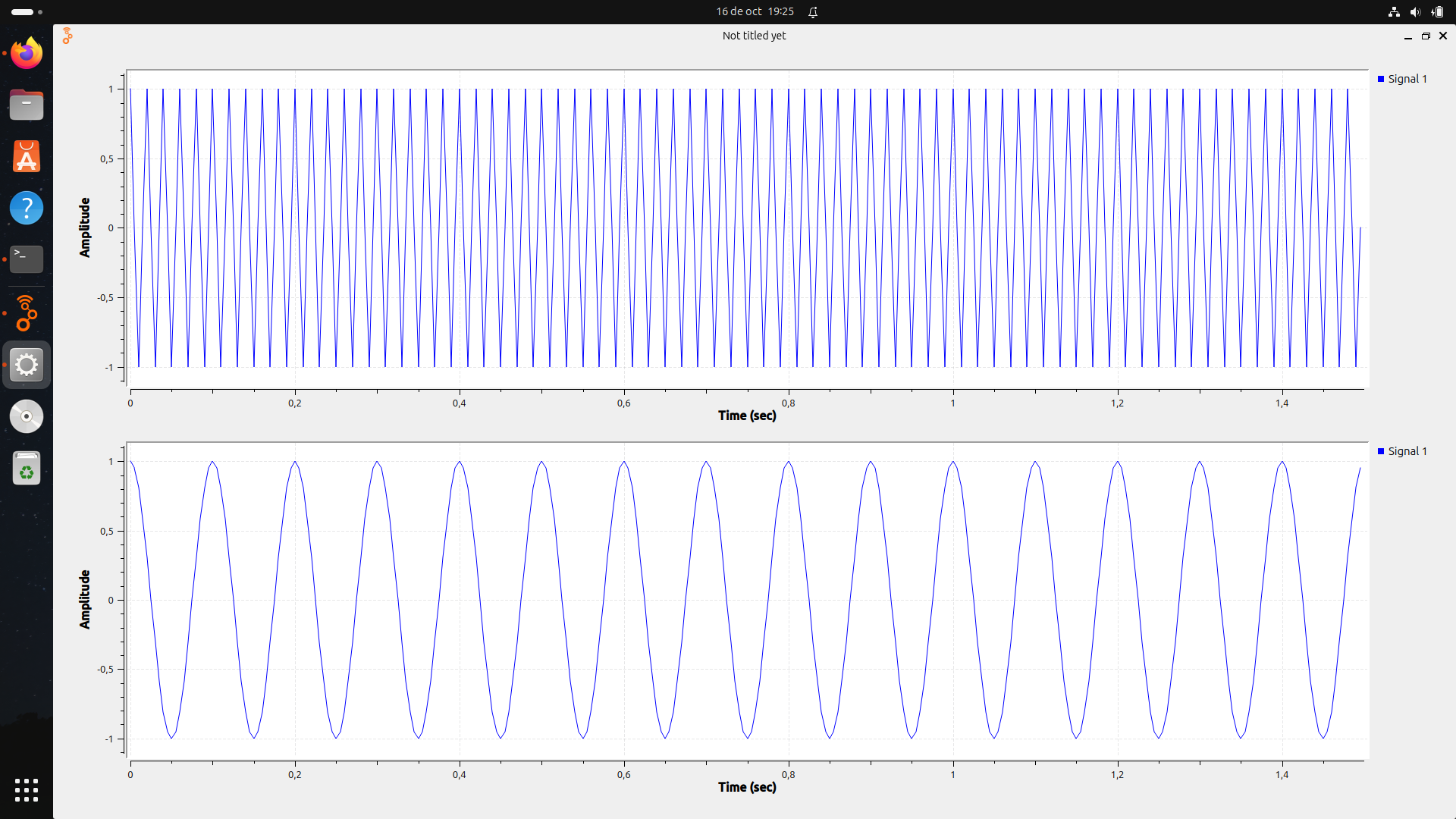




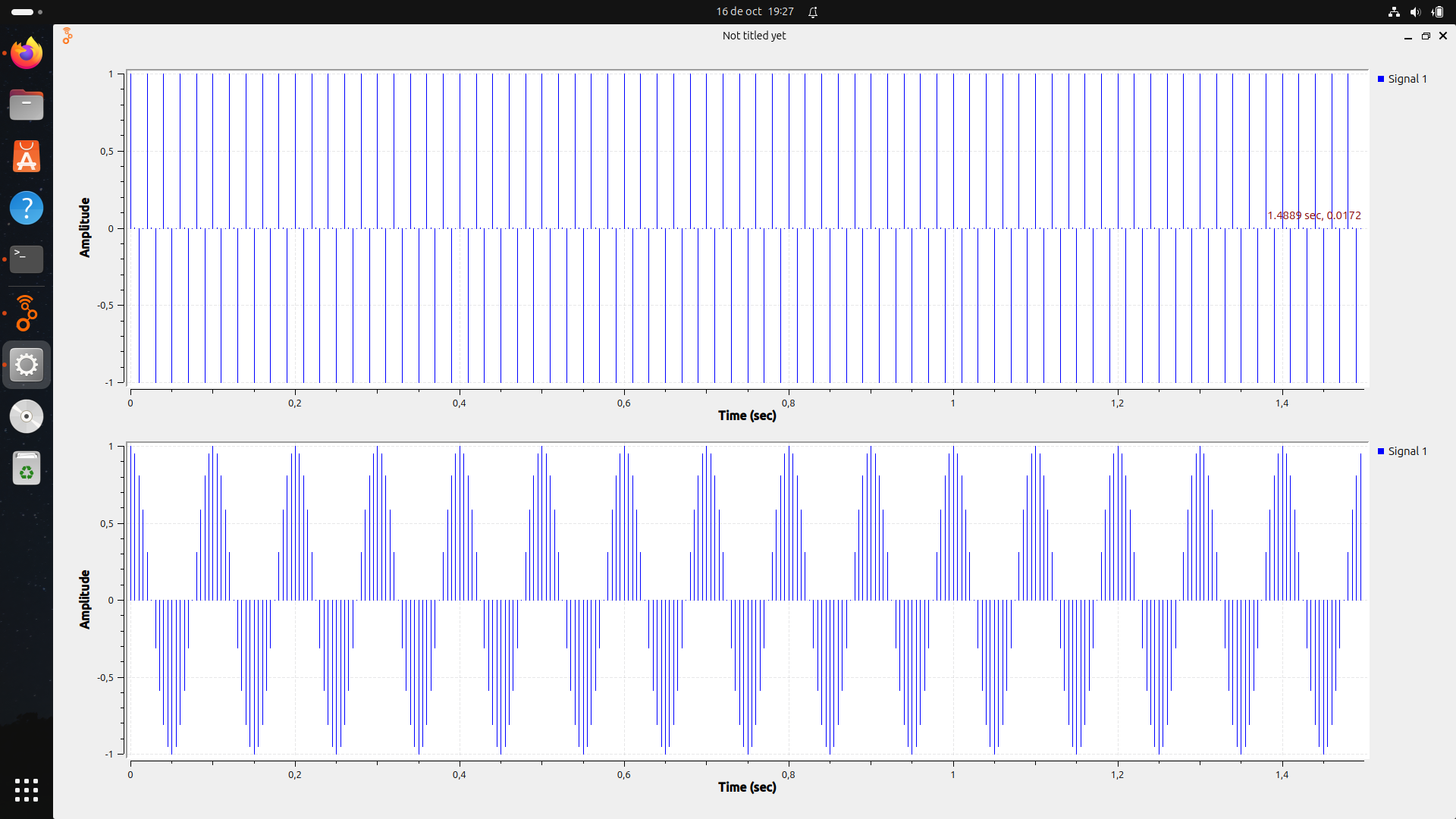
Aplicando stem plot:



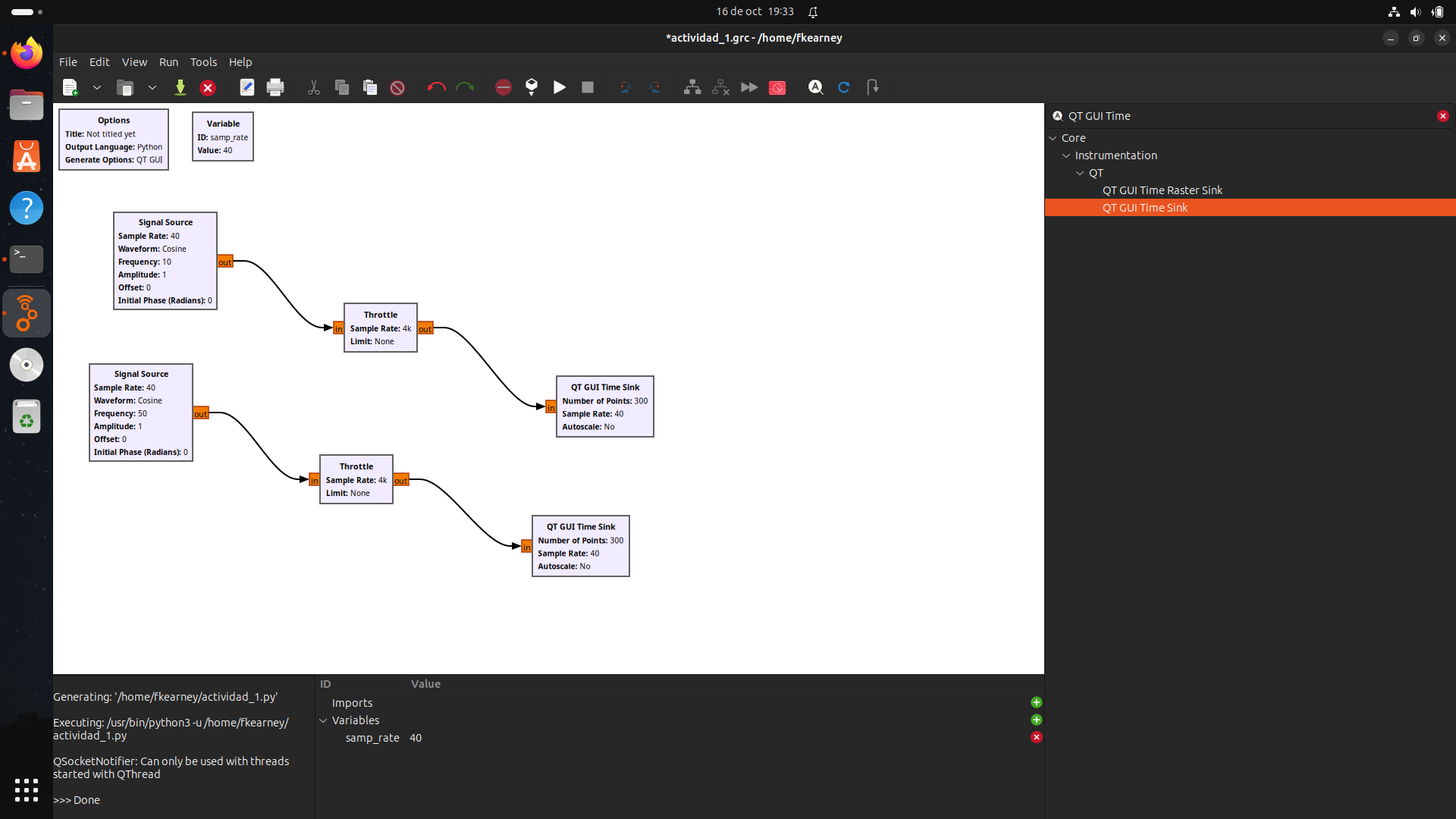
-Señal 2():

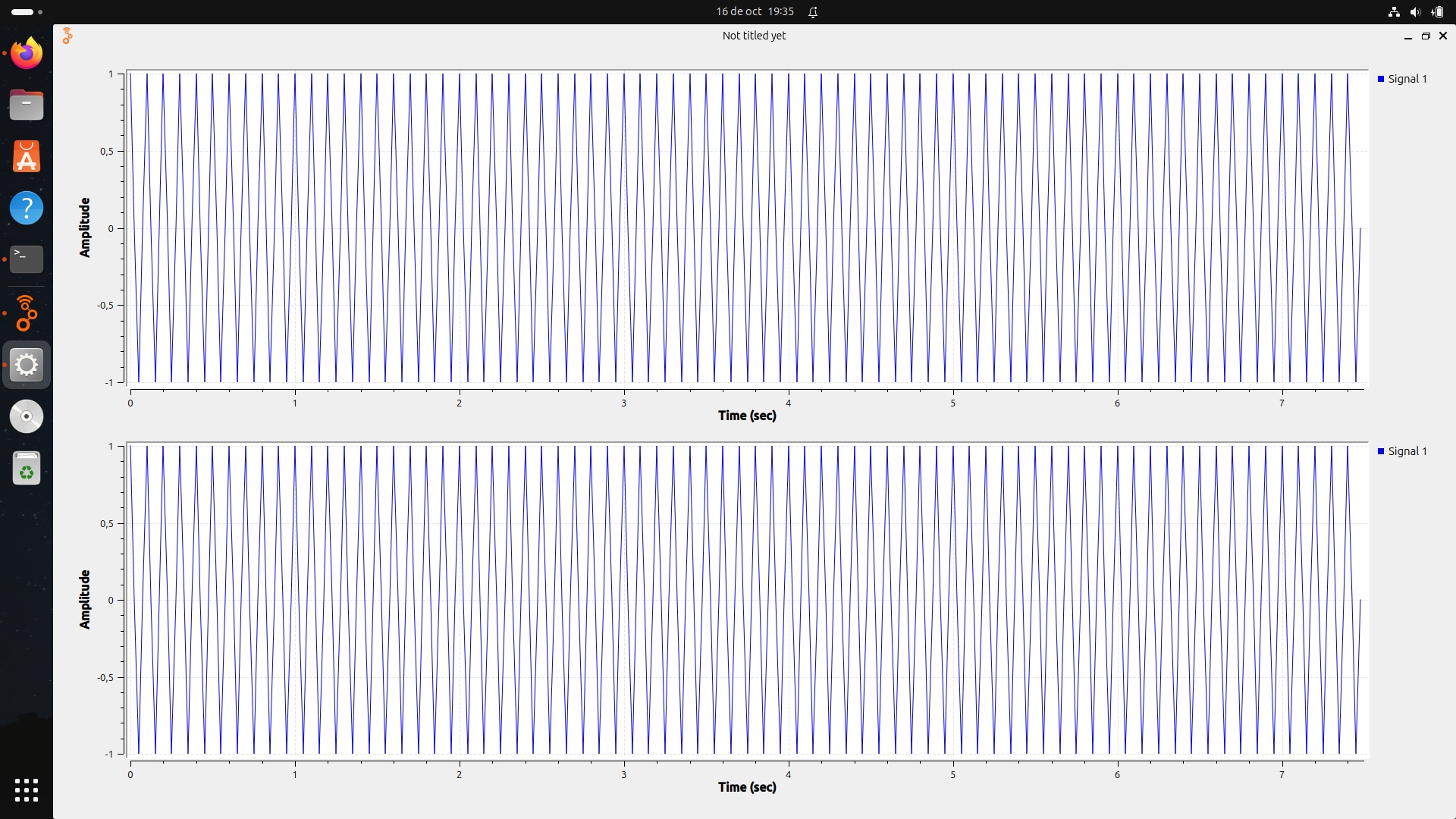


Aplicando stem plot:



Aplicando una tasa de muestreo fs, a 40 muestras por segundo (40 Hertz):





Aplicando stem plot:

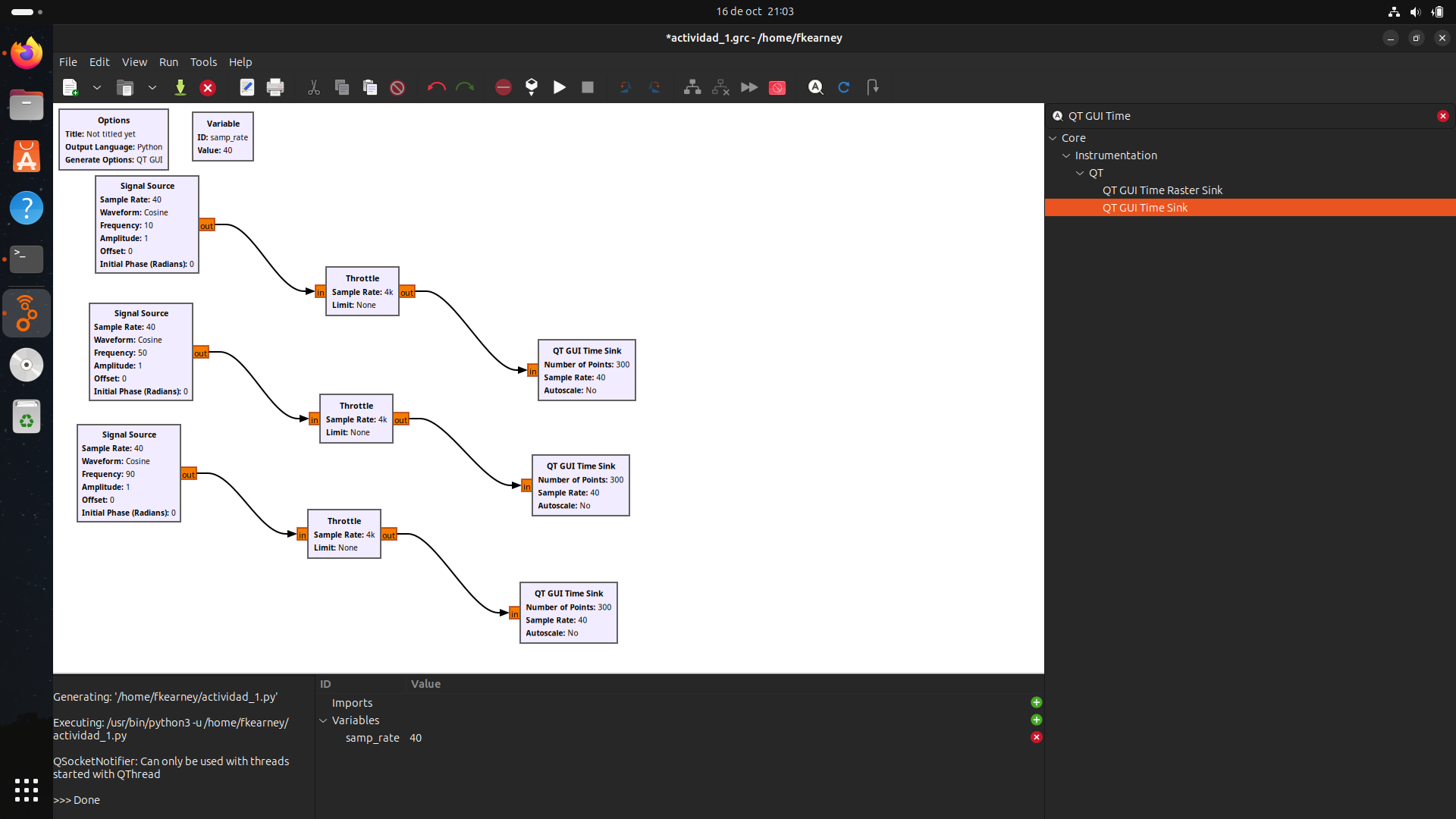


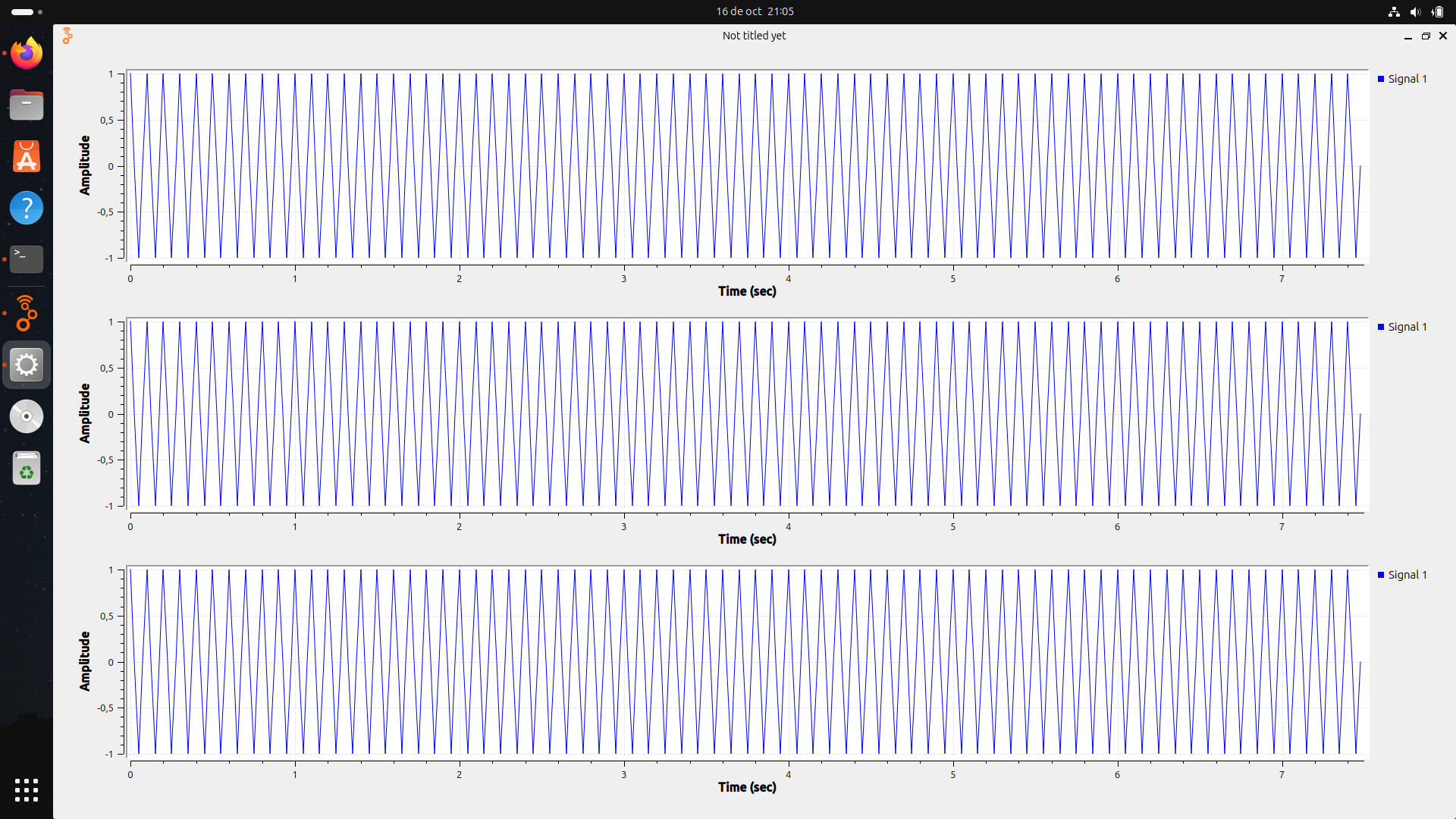
1. El fenómeno de aliasing que se observa en el inciso b se debe a que por haber muestreado las dos señales con una tasa de muestreo fs=40Hz, eso nos generó un alias. Generó que la termine siendo un alias de la . Este alias justamente, cuando lo graficamos hace que la sea la misma señal que la . Esto se puede explicar mediante la siguiente relación:

1. Las señales que producen un alias de la señal [n] cuando fs=40 Hertz pueden ser todas aquellas que cumplan con la condición:

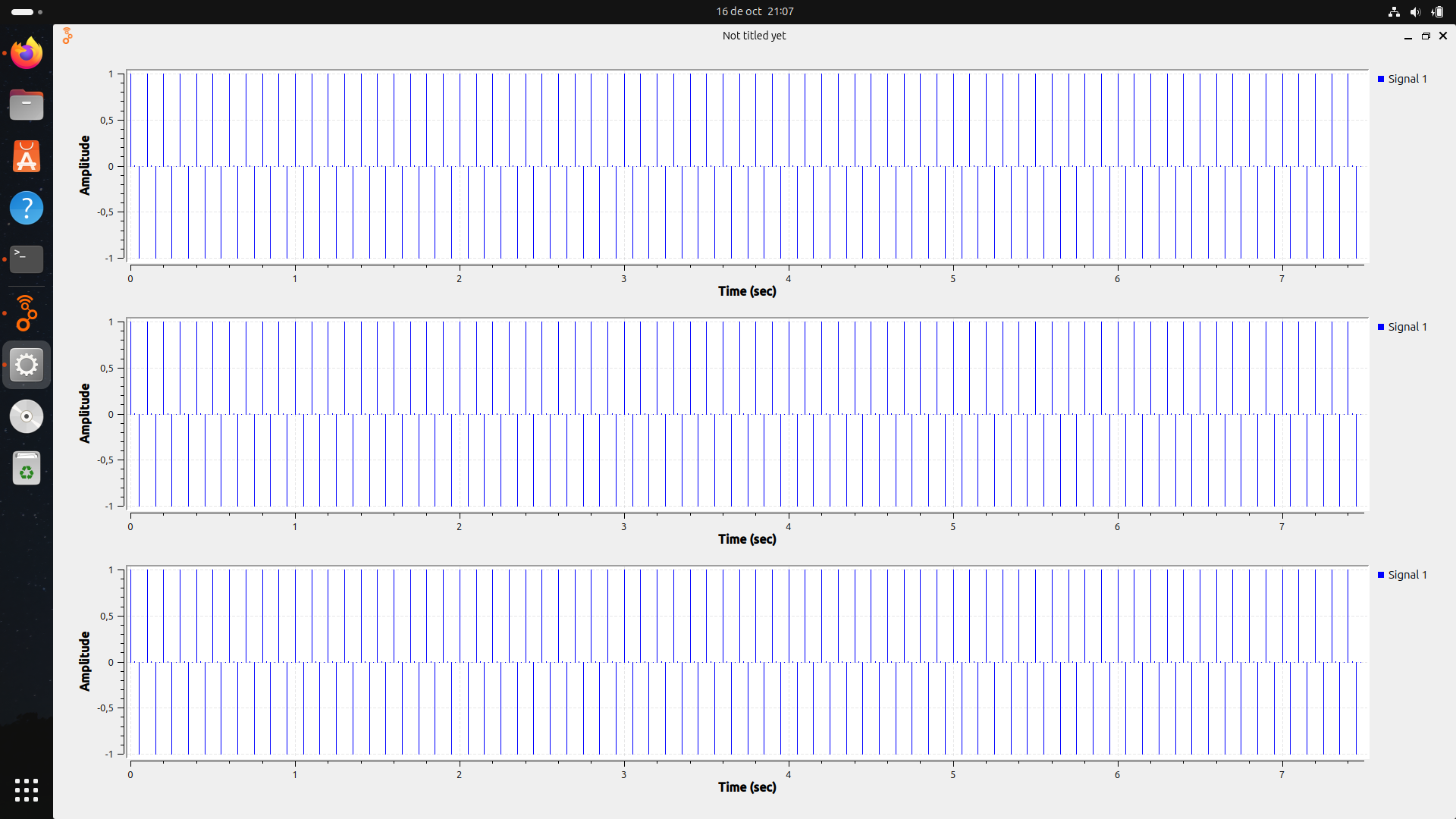
Un ejemplo en GNU Radio Companion puede ser tomando un k=2, por lo que:

→ →





Aplicando stem plot:



1. La ecuación matemática para determinar todas las señales tales que, al ser muestreadas con fs=40 Hertz, son indistinguibles de la señal x1[n] cuando ha sido muestreada también a esta fs es la siguiente:

**Ejercicio 3:**

**Resolución:**

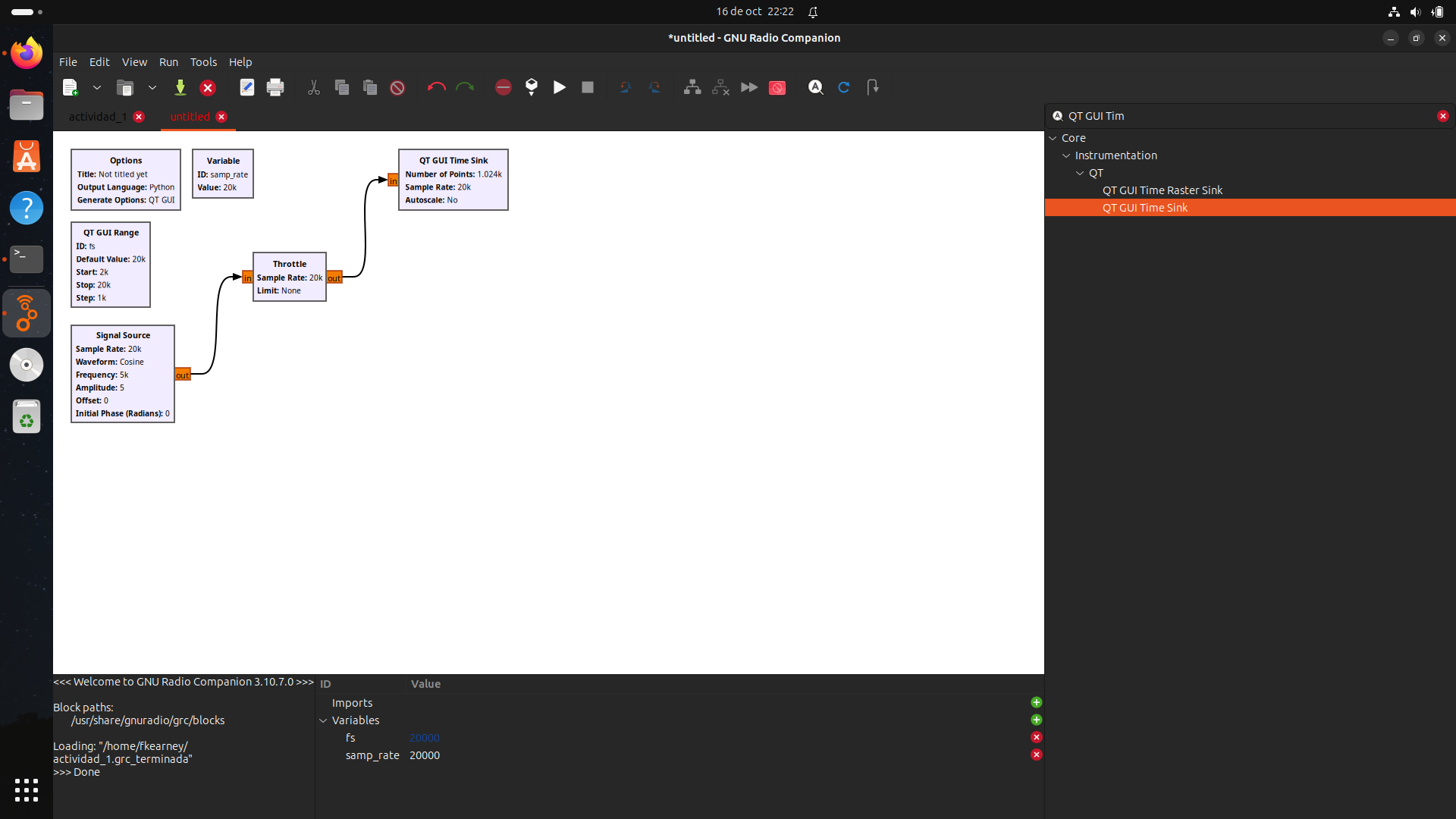
;

—>

Frecuencia de muestreo inicial:

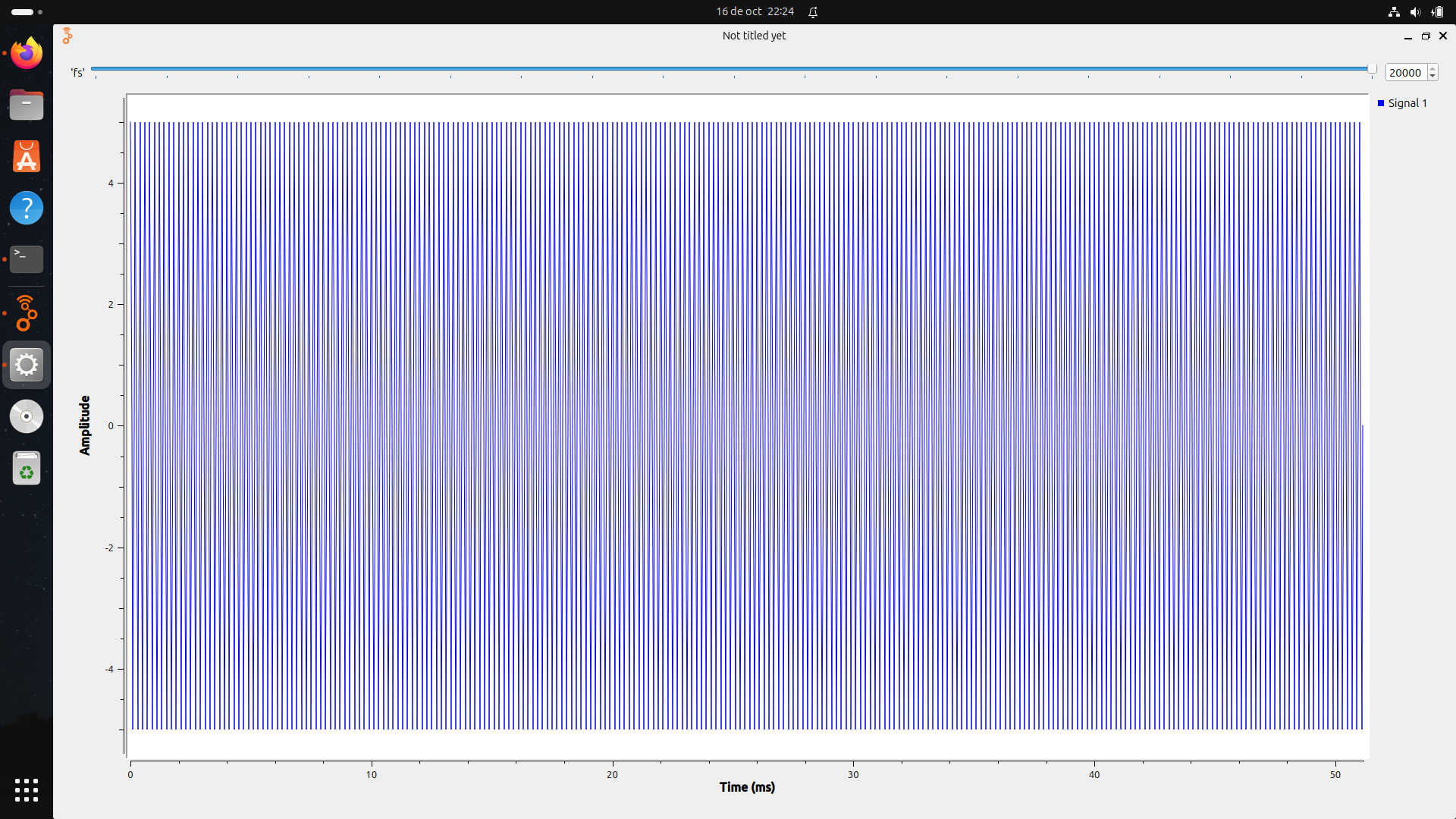
—> Consideraremos inicialmente una

Caso inicial:

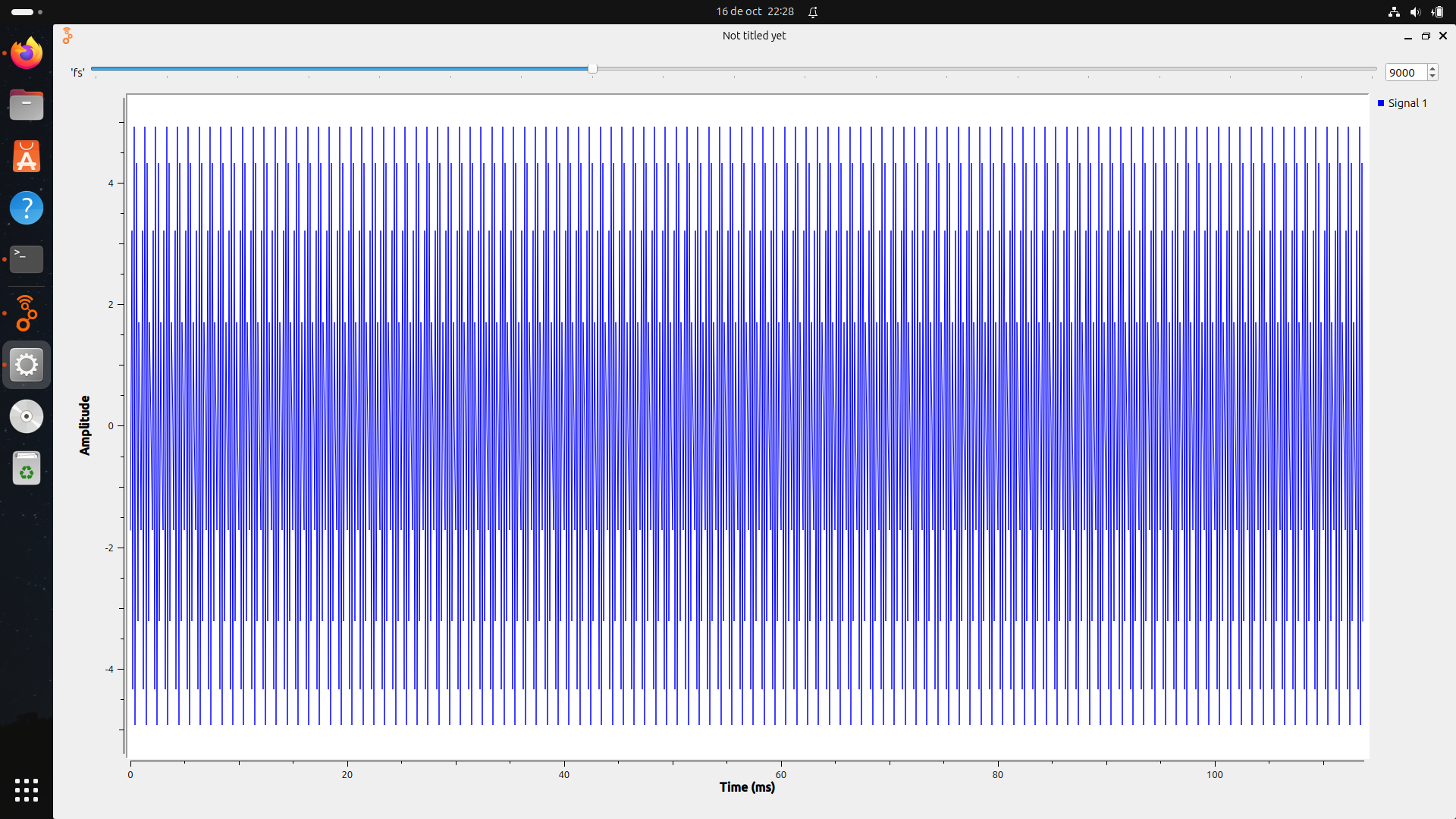
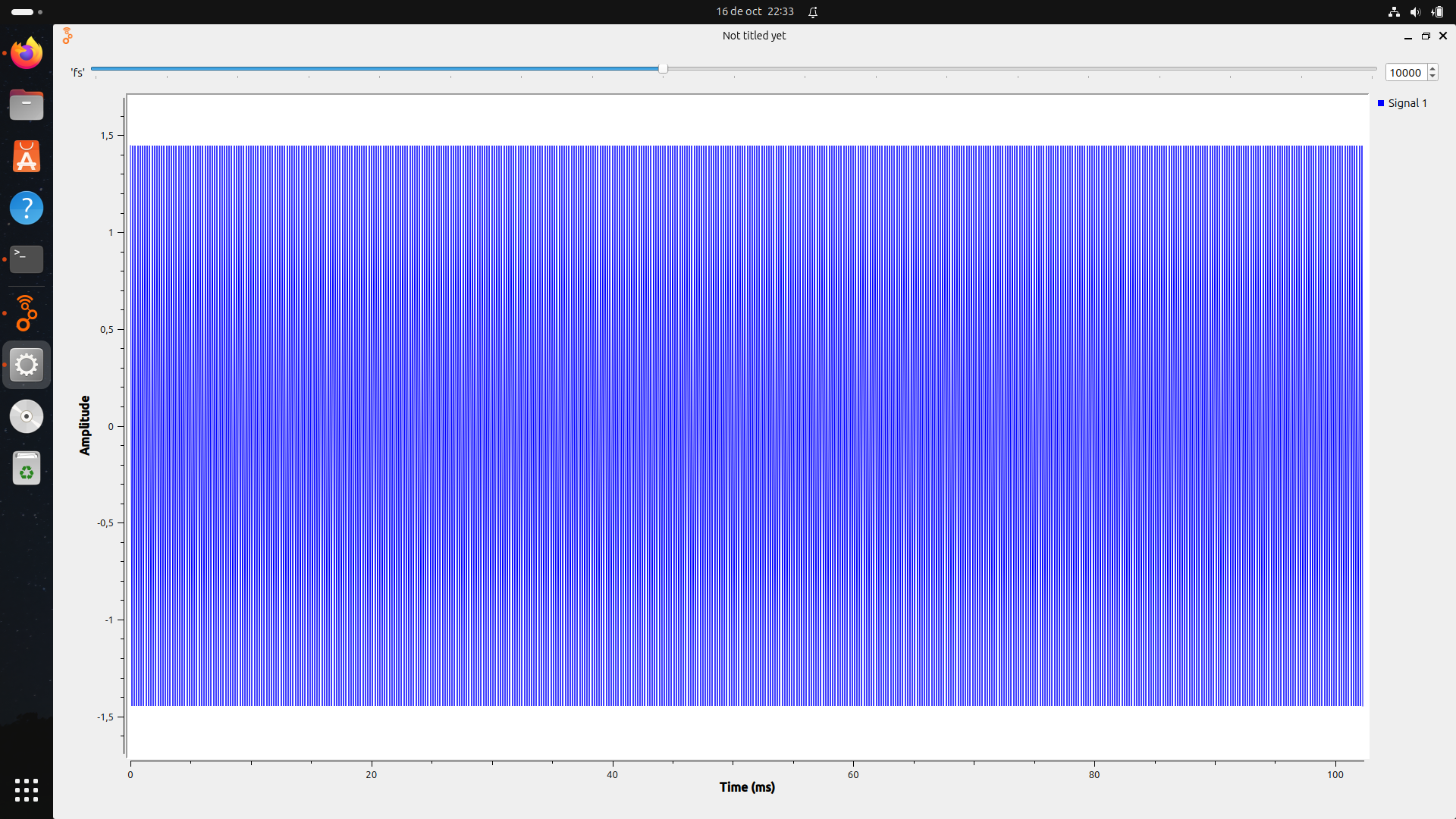


1. Cuando usamos una fs menor a la mínima requerida por el criterio de Nyquist aparece aliasing, la señal parece tener otra frecuencia. A continuación lo veremos gráficamente:

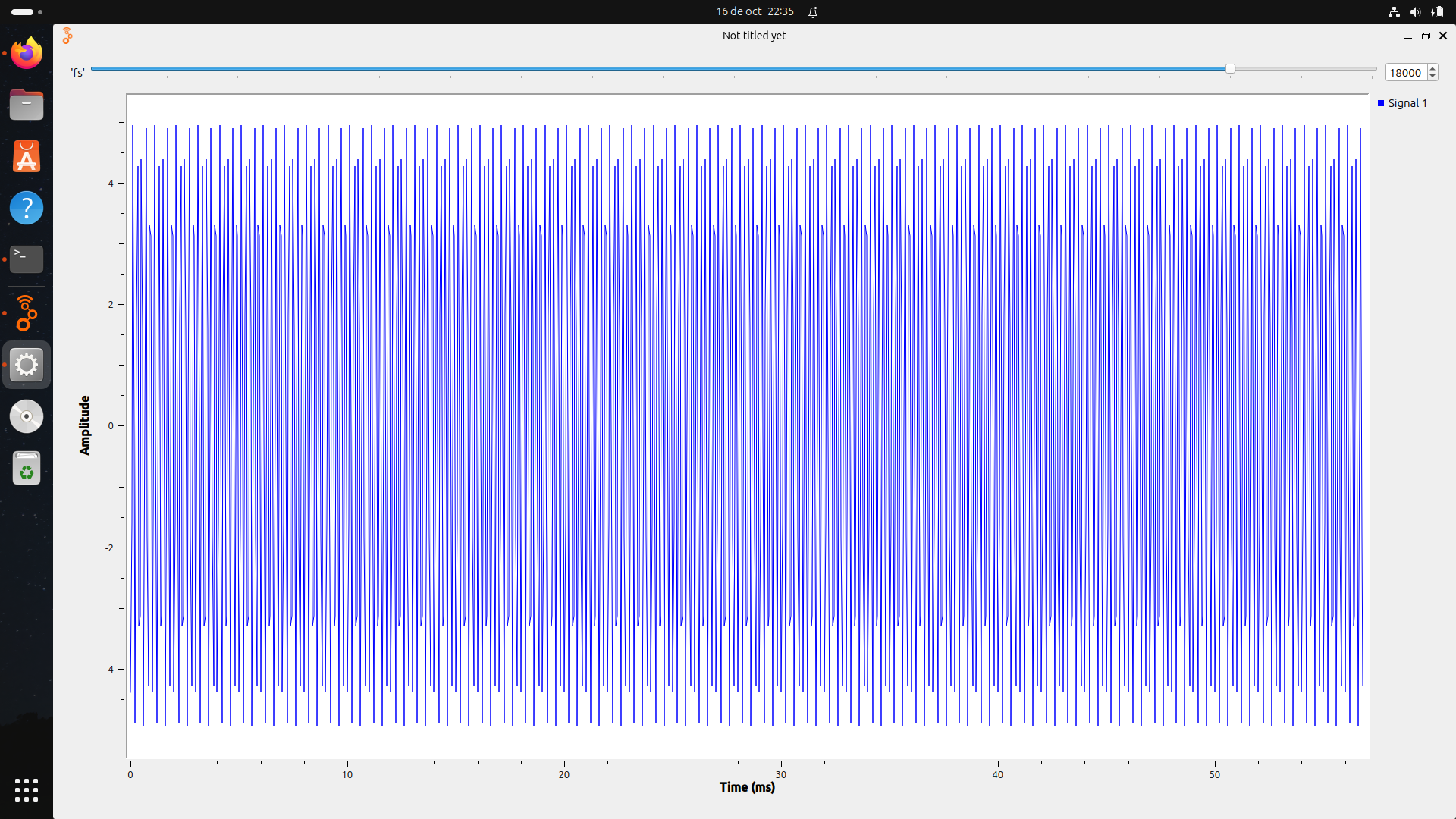
Nos movemos a una fs menor a la mínima(9000 Hz)



1. Una fs igual a la de Nyquist produce que se cumpla Nyquist, pero solo hay 2 muestras por ciclo → la señal se ve mínima.

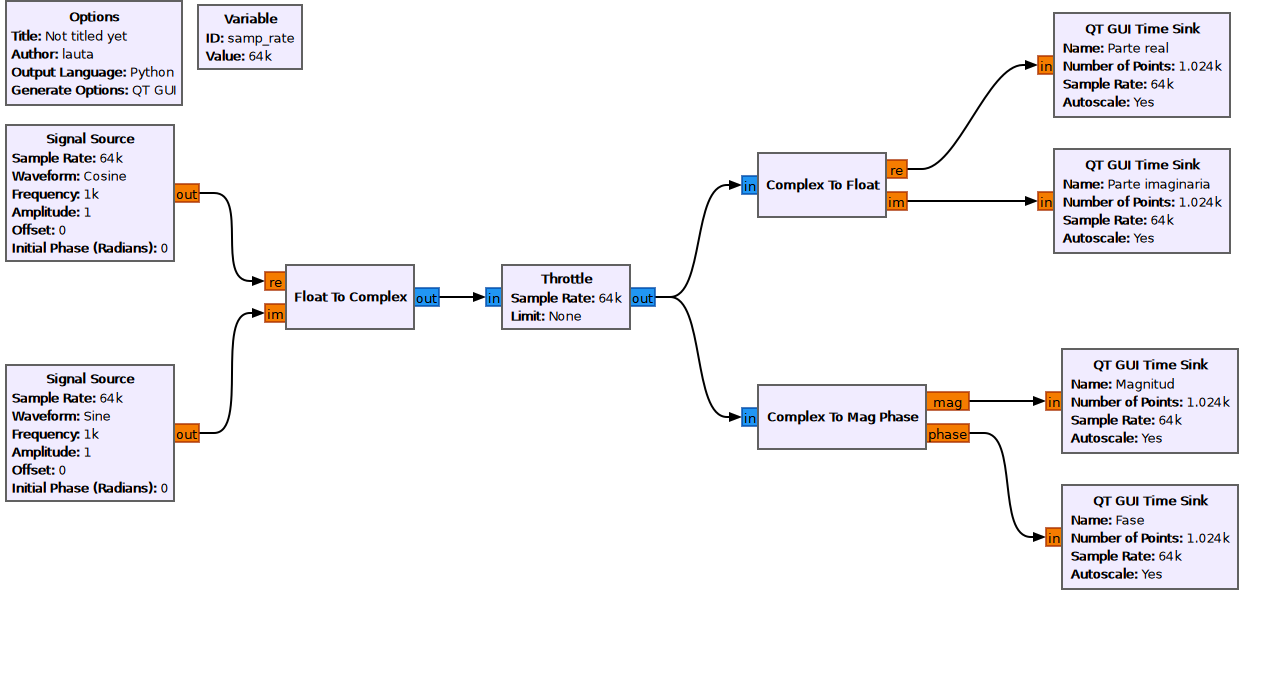


1. Una fs superior a la de Nyquist produce que la señal se reconstruya bien, y al aumentar fs se ve más suave.



**Ejercicio 5**:

Representación Fasorial de x(t):

La señal compleja representa un fasor que gira en el plano complejo. La velocidad angular del fasor está determinada por el argumento de la exponencial compleja, resultando en una velocidad angular de 2000π rad/s. En términos de movimiento rotacional, el fasor completa 1000 vueltas completas durante un intervalo de un segundo, lo que corresponde a su frecuencia fundamental de 1000 Hz. El tiempo requerido para que el fasor complete una vuelta completa es de 1 milisegundo, que equivale al período fundamental de la señal.

## 

## 

## Representación en Coordenadas Ortogonales: Parte Real e Imaginaria

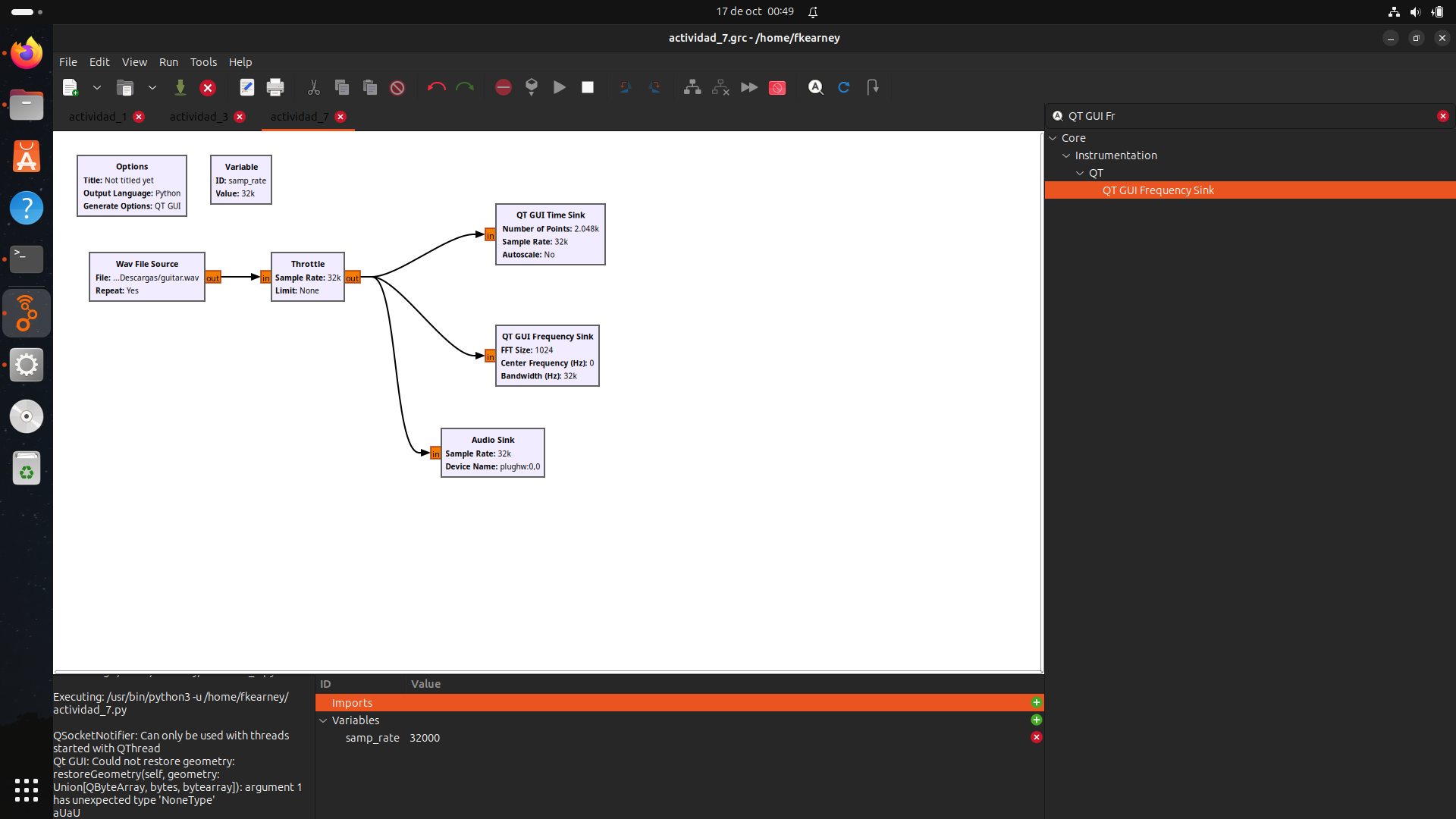
Para visualizar las componentes real e imaginaria de la señal, se implementó en GNU Radio una configuración que genera la señal compleja mediante la combinación de dos señales reales ortogonales. La parte real de la señal corresponde a la función cos(2π×1000t), que se visualiza como una onda cosenoidal de 1 kHz con amplitud unitaria. La parte imaginaria corresponde a la función sin(2π×1000t), que se manifiesta como una onda senoidal de la misma frecuencia y amplitud, pero desfasada 90 grados respecto a la componente real. Estas dos señales representan las proyecciones del fasor rotatorio sobre los ejes real e imaginario del plano complejo.

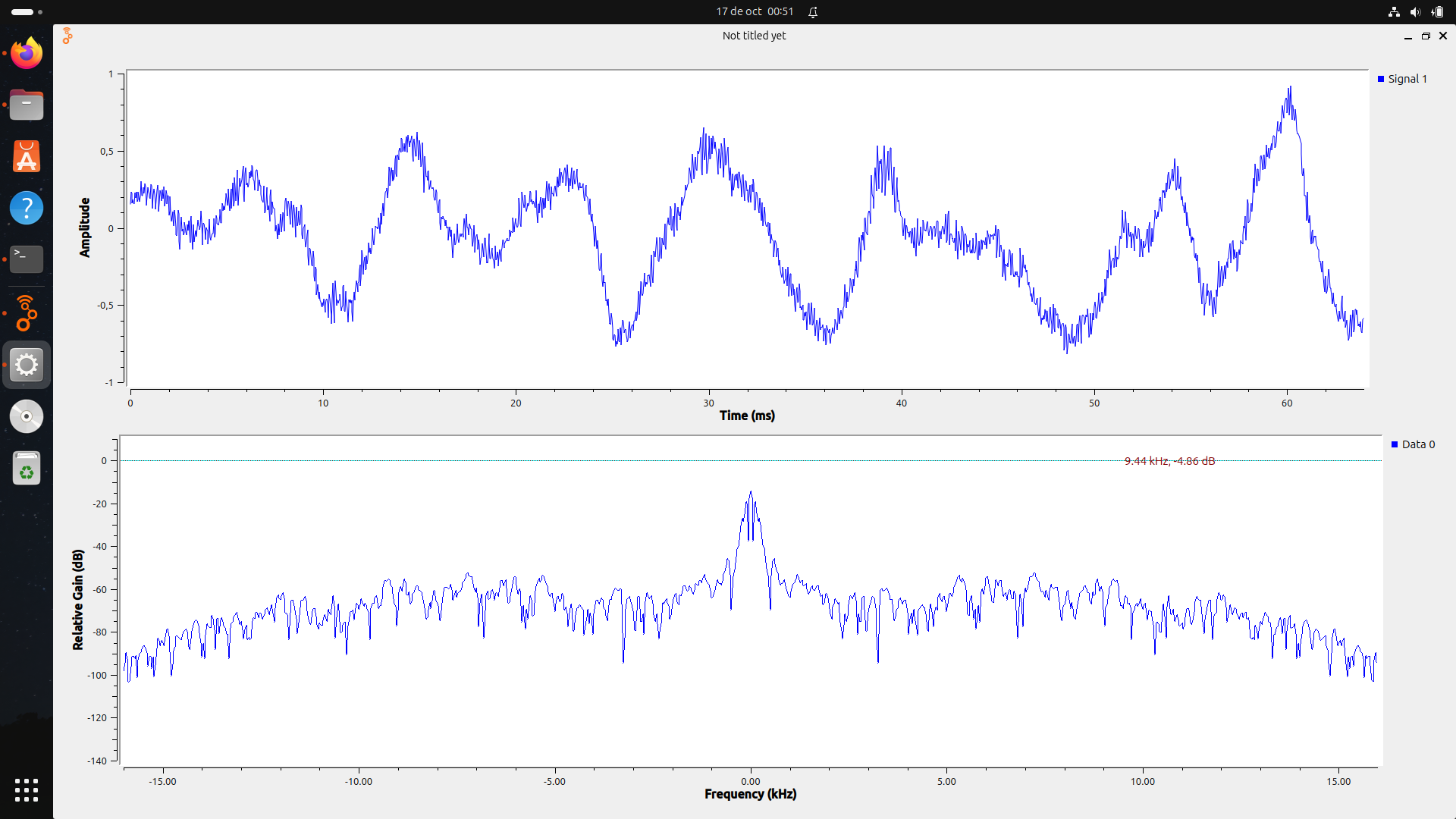
## Representación en Coordenadas Polares: Magnitud y Fase

El análisis en coordenadas polares revela que la magnitud de la señal permanece constante en un valor de 1 para todo instante de tiempo, confirmando que el fasor se desplaza sobre la circunferencia unitaria del plano complejo. La fase de la señal exhibe un comportamiento lineal creciente con el tiempo, describiendo una rampa que se repite periódicamente cada 2π radianes. A los 10.30 milisegundos, el valor de la fase es de 1.885 radianes, equivalente a aproximadamente 108 grados. En este instante específico, el fasor ha completado 10.3 vueltas completas alrededor del origen y se posiciona en el segundo cuadrante del plano complejo, con coordenadas aproximadas de (-0.309, 0.951) en su representación rectangular.

La implementación práctica en GNU Radio permitió verificar el comportamiento teórico de la señal compleja, confirmando las relaciones matemáticas entre sus diferentes representaciones y validando las propiedades fundamentales de las señales exponenciales complejas en el dominio del tiempo.

**Ejercicio 7**:





1. En el QT GUI Time Sink vemos una onda casi sinusoidal, pero con pequeñas irregularidades. Eso se debe a los armónicos naturales de la cuerda de guitarra.

La frecuencia que podemos ver es de un pico principal alrededor de 440 Hz, que corresponde al tono fundamental. También aparecen picos más pequeños en múltiplos de 440 Hz (880, 1320, etc.), que son los armónicos.  
Por lo tanto:

1. La nota musical correspondiente a este tono es la A4(La4) en afinación estándar que corresponde a una frecuencia de 440Hz.

En el sistema latino es la A4=La4, y en el cifrado americano es la A4=440Hz

**Ejercicio 9**:

**a. Representación de Muestras y Error de Cuantización**

La representación de muestras utilizando variables de punto flotante de 32 bits (float32) introduce un error de cuantización extremadamente pequeño, pero no nulo. Esto se debe a que el formato float32 tiene una precisión limitada por el número de bits disponibles para la mantisa, lo que resulta en una cuantización no uniforme across el rango dinámico. Sin embargo, para la mayoría de aplicaciones prácticas en procesamiento de señales, este error es despreciable debido a la alta precisión del formato.

No es posible evitar completamente el error de cuantización cuando trabajamos con señales digitales, ya que toda representación digital implica la discretización de valores en un conjunto finito de niveles. La única manera de reducir el error es aumentar el número de bits de representación o emplear técnicas avanzadas como el dithering, pero la eliminación total del error de cuantización no es achievable en sistemas digitales.

**b. Cuantización a 12 Bits**

Al implementar en GNU Radio Companion una señal sinusoidal de 100 Hz con frecuencia de muestreo de 10 kHz y amplitud 1, cuantizada a 12 bits, se observa que las diferencias entre la señal original y la señal cuantizada son mínimas y visualmente imperceptibles en las gráficas temporales. Esto demuestra que con una cuantización de 12 bits, la resolución es suficientemente alta para representar fielmente la señal original.

Para 12 bits, el número de niveles de cuantización disponibles es 4,096 (2¹²), lo que proporciona una resolución muy fina para la señal sinusoidal.

**c. Cuantización a 4 Niveles**

Para lograr 4 niveles de cuantización, se necesitan 2 bits (2² = 4). Al comparar gráficamente la señal original sin cuantizar con la señal cuantizada a 4 niveles, se observan diferencias significativas en la forma de onda. La señal cuantizada muestra una característica escalonada con saltos discretos entre los cuatro niveles disponibles, en contraste con la curva suave de la señal sinusoidal original.

**d. Cálculo Teórico de SQNR**

Para una señal sinusoidal con cuantización de 4 niveles (2 bits), la relación señal-ruido de cuantización (SQNR) teórica se calcula como:

SQNR\_{dB} = 6.02 × n + 1.76 dB  
SQNR\_{dB} = 6.02 × 2 + 1.76 = 13.8 dB

Por cada bit adicional que se agrega al cuantizador, la SQNR aumenta aproximadamente 6 dB. Esta mejora se debe a que cada bit adicional cuadruplica el número de niveles de cuantización, reduciendo el error de cuantización por un factor de 4, lo que corresponde a una mejora de 6 dB en la relación señal-ruido.

**e. Estimación del Error de Cuantización**

Al estimar el error de cuantización e\_q[n] para una señal sinusoidal con cuantización de 4 niveles, se observa en el gráfico temporal un patrón periódico que sigue la forma de la señal original. El error muestra una estructura característica con transiciones bruscas en los puntos donde la señal cuantizada cambia entre niveles discretos. La potencia de este error puede calcularse mediante el bloque de potencia media.

**f. Implementación del Bloque de Potencia Media y Cálculo de SQNR**

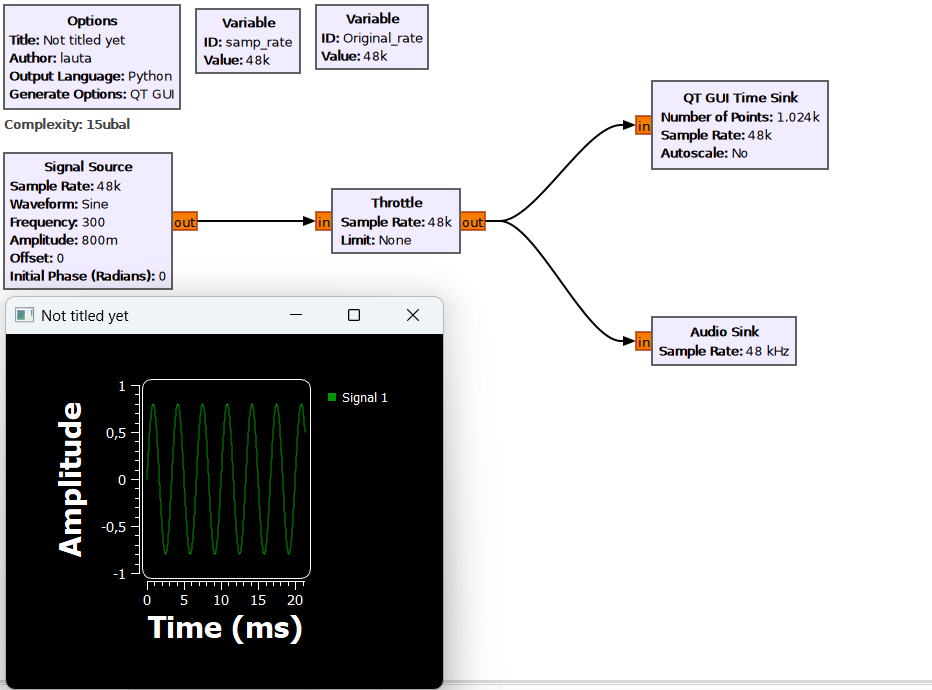
i. Gráficas de P\_x[n] y P\_q[n]: Al implementar el bloque "Potencia media" en GNU Radio Companion, las gráficas de P\_x[n] (potencia de la señal original) y P\_q[n] (potencia del error de cuantización) muestran valores constantes en el tiempo, confirmando que se trata de señales estacionarias. P\_x[n] se mantiene en aproximadamente 0.5, mientras que P\_q[n] muestra un valor significativamente menor.

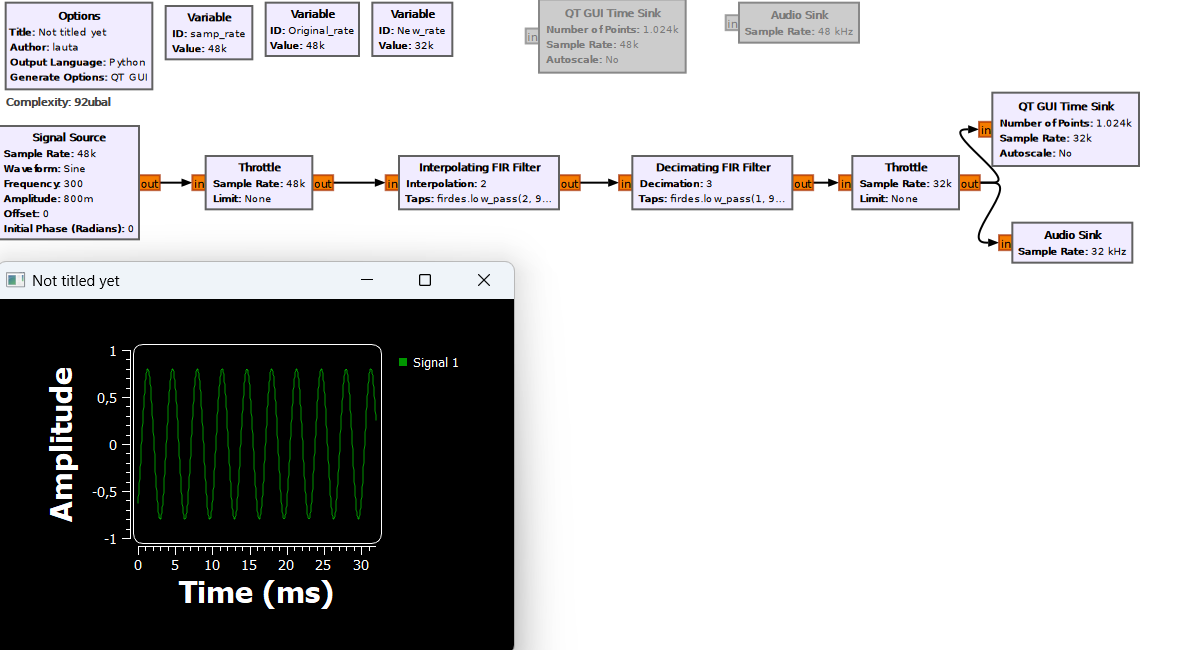
ii. Cálculo y Gráfica de SQNR: La relación SQNR calculada en Companion se obtiene mediante la división de P\_x entre P\_q, seguida de una conversión a escala logarítmica (dB). La gráfica resultante muestra un valor constante de SQNR que puede compararse con el cálculo teórico.

iii. Comparación con el Valor Teórico: Las diferencias entre el valor de SQNR teórico (13.8 dB) y el obtenido en GNU Radio Companion pueden atribuirse a varios factores, la señal generada no es una sinusoidal perfecta debido a limitaciones numéricas, el cálculo de potencia media se realiza sobre un número finito de muestras, la cuantización implementada puede tener características no ideales, los bloques utilizados introducen retardos y efectos transitorios no considerados en el análisis teórico

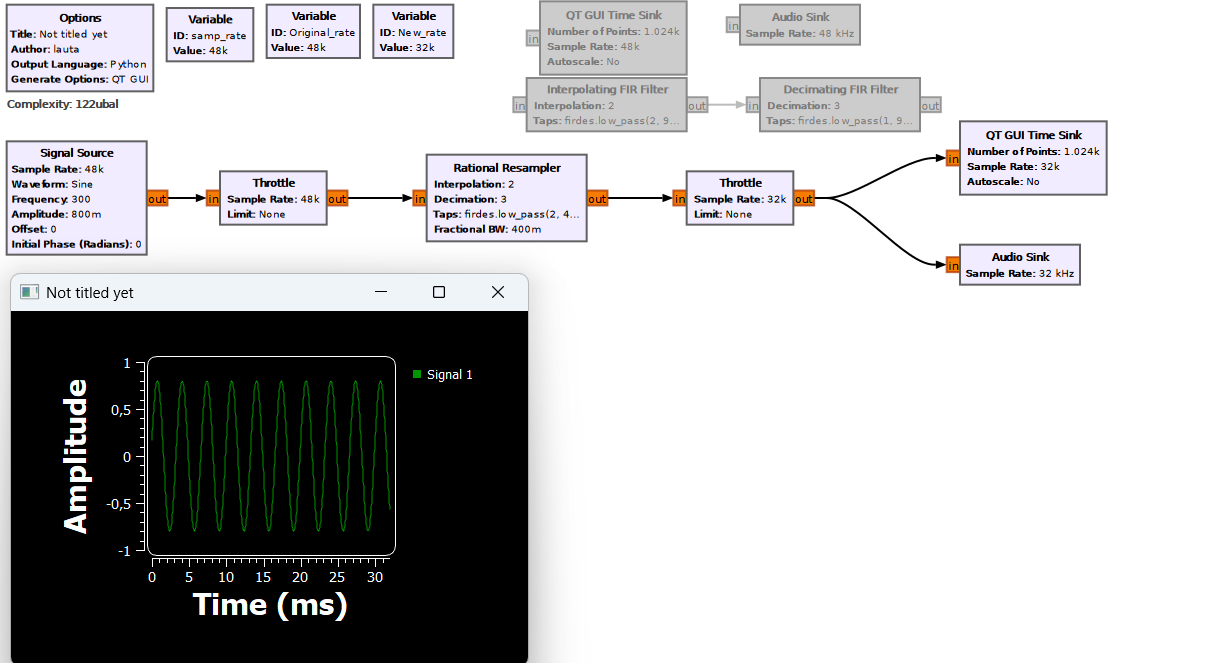
La implementación práctica en GNU Radio Companion valida los conceptos teóricos de cuantización y demuestra cómo la relación señal-ruido de cuantización depende críticamente del número de bits utilizados en la representación digital de las señales.

**Ejercicio 11**:

Se generó una señal sinusoidal de 300 Hz con una tasa de muestreo de 48 kHz utilizando el bloque Signal Source en GNU Radio Companion. La señal original se reprodujo exitosamente mediante el bloque Audio Sink y se visualizó con QT GUI Time Sink, confirmando una forma de onda sinusoidal pura con un período de aproximadamente 160 muestras, equivalente a la frecuencia esperada de 300 Hz. Esta configuración inicial sirvió como referencia para posteriormente implementar el proceso de conversión de tasa de muestreo.

Para convertir la señal a 32 kHz, se implementó inicialmente un método combinado de interpolación y diezmado. La interpolación se realizó primero con un factor de 2, elevando temporalmente la tasa de muestreo a 96 kHz mediante un Interpolating FIR Filter configurado con un filtro paso bajo. Seguidamente, se aplicó un diezmado con factor de 3 a través de un Decimating FIR Filter, logrando la tasa objetivo de 32 kHz. Este orden de operaciones —interpolación seguida de diezmado— se eligió estratégicamente para prevenir el aliasing y preservar la calidad de la señal, ya que al interpolar primero se incrementa la frecuencia de Nyquist temporal, permitiendo un filtrado anti-aliasing más efectivo antes de la reducción de muestras.

La verificación gráfica demostró que la señal convertida mantenía la frecuencia fundamental de 300 Hz pero con un período de aproximadamente 107 muestras, consistente con la nueva tasa de muestreo de 32 kHz. La forma de onda sinusoidal se preservó sin distorsiones evidentes, validando la efectividad del método.

Posteriormente, se reemplazó esta implementación por un bloque Rational Resampler configurado con interpolación de 2, diezmado de 3 y un Fractional BW de 0.4. Este abordaje unificado simplificó significativamente el flujograma al combinar ambas operaciones en un solo bloque optimizado. El parámetro Fractional BW de 0.4 permitió un balance adecuado entre la atenuación del aliasing y la respuesta temporal del filtro interno. La reproducción de la señal resultante mediante Audio Sink confirmó que el tono de 300 Hz se mantenía sin distorsión audible, mientras que la visualización temporal mostró una señal sinusoidal limpia y consistente.

Al comparar ambos métodos, se observó que el enfoque de bloques separados proporciona un control más granular sobre cada etapa del proceso y resulta valioso para fines educativos, pero introduce mayor complejidad en el diseño. En contraste, el Rational Resampler ofrece una implementación más compacta y eficiente, con optimizaciones internas que reducen la carga computacional. Ambos métodos produjeron resultados equivalentes en términos de calidad de señal y precisión en la frecuencia fundamental, pero el Rational Resampler demostró ser más práctico para aplicaciones reales debido a su simplicidad y eficiencia. La elección entre métodos depende entonces de los requisitos específicos de control versus optimización en cada aplicación particular de procesamiento de señales.

**Ejercicio 12**:

**a. Cálculo de Tasa de Bits por Segundo**

La tasa de bits o bitstream en un sistema RTL-SDR se determina considerando los parámetros fundamentales de adquisición de señales. Para una frecuencia de muestreo de 1.2 MHz, trabajando con los dos canales I y Q (In-phase y Quadrature), y considerando que la RTL-SDR utiliza típicamente una resolución de 8 bits por muestra, el cálculo de la tasa de bits se realiza de la siguiente manera:

La fórmula general para calcular la tasa de bits es:  
Tasa de bits = Frecuencia de muestreo × Bits por muestra × Número de canales

Sustituyendo los valores conocidos:

* Frecuencia de muestreo: 1,200,000 muestras/segundo
* Bits por muestra: 8 bits
* Número de canales: 2 (I y Q)

Tasa de bits = 1,200,000 × 8 × 2 = 19,200,000 bps = 19.2 Mbps

Esta tasa de 19.2 Mbps representa el flujo de datos brutos que genera el dispositivo cuando opera a 1.2 MHz de frecuencia de muestreo, lo que debe considerarse para el dimensionamiento de sistemas de almacenamiento y procesamiento.

**b. Niveles de Cuantización de las Muestras**

La cuantización es el proceso mediante el cual las señales analógicas continuas se convierten en valores digitales discretos. En el caso de la RTL-SDR, la resolución del conversor analógico-digital (ADC) determina el número de niveles de cuantización disponibles para representar las señales capturadas.

Las RTL-SDR estándar utilizan un ADC de 8 bits, por lo que el número de niveles de cuantización se calcula como:  
Niveles de cuantización = 2^(bits) = 2⁸ = 256 niveles

Estos 256 niveles se distribuyen uniformemente a lo largo del rango dinámico del ADC, permitiendo representar las amplitudes de las señales I y Q con una resolución de 8 bits. Cada muestra I y cada muestra Q puede tomar cualquiera de estos 256 valores discretos, proporcionando una representación digital de la señal analógica original.

**c. Frecuencia de Muestreo Máxima Soportada**

La frecuencia de muestreo máxima de las RTL-SDR está determinada por las limitaciones hardware del dispositivo, particularmente del chip RTL2832U y la interfaz USB. Según las especificaciones técnicas de los modelos más comunes disponibles en el mercado:

RTL-SDR Blog V3 (modelo ampliamente utilizado):

* Frecuencia de muestreo máxima: 2.4 MHz
* Frecuencia de muestreo óptima recomendada: 2.048 MHz

Modelos genéricos de RTL-SDR:

* Frecuencia de muestreo máxima teórica: 3.2 MHz
* Frecuencia de muestreo práctica estable: 2.4 MHz

La limitación principal proviene del ancho de banda disponible en la interfaz USB 2.0, que debe transportar el flujo de datos de ambos canales I y Q. A 2.4 MHz de frecuencia de muestreo, la tasa de datos calculada es:  
2,400,000 × 8 × 2 = 38.4 Mbps

Esta tasa se encuentra dentro de las capacidades típicas de USB 2.0, que puede manejar hasta aproximadamente 280-320 Mbps en condiciones ideales, pero considerando overhead del protocolo y otros factores, 2.4 MHz representa un equilibrio óptimo entre ancho de banda y estabilidad.

Es importante destacar que operar a frecuencias de muestreo superiores a 2.4 MHz puede resultar en pérdida de paquetes USB, artefactos en las señales y comportamiento inestable del dispositivo, por lo que se recomienda mantenerse en 2.4 MHz o inferior para aplicaciones que requieran confiabilidad en la adquisición de datos.

**Conclusión:**

Para concluir nuestro trabajo práctico desarrollado con GNU Radio Companion en el marco de la asignatura Sistemas y Señales II de la Universidad Nacional de Río Cuarto en el año 2025, podemos decir que nos ha permitido una profunda validación experimental de los principios teóricos fundamentales del procesamiento digital de señales.

Consideramos que este conjunto de ejercicios con GNU Radio Companion nos ha facilitado la comprensión de los conceptos claves de la teoría de señales, incluyendo el muestreo, el aliasing, la cuantización y la conversión de tasas. La guía de trabajos nos demuestra la capacidad de implementar y verificar sistemas de procesamiento digitales de señales de manera práctica, estableciendo una base empírica sólida para futuros estudiantes y también ayudando en la implementación de aplicaciones para nuestra querida carrera, Ingeniería en Telecomunicaciones.