

UNIVERSIDAD POLITÉCNICA DE VICTORIA

Nombre del Reporte:

Graficación de señales continuas y discretas materiales

Que presenta el alumno:

De la Cruz Rivera Yan Carlo Guadalupe

De la carrera de:

Ingeniería en Mecatrónica

Asignatura:

Adquisición y Procesamiento Digital de Señales

Impartida por:

M.C. CARLOS ANTONIO TOVAR GARCIA

Cd. Victoria, Tamaulipas, Mayo de 2025

Objetivos y descripción de señales

El objetivo de esta tarea es generar y visualizar diferentes tipos de señales tanto en forma continua como discreta, para entender su comportamiento y cómo se representan digitalmente.

Las señales consideradas en esta tarea son:

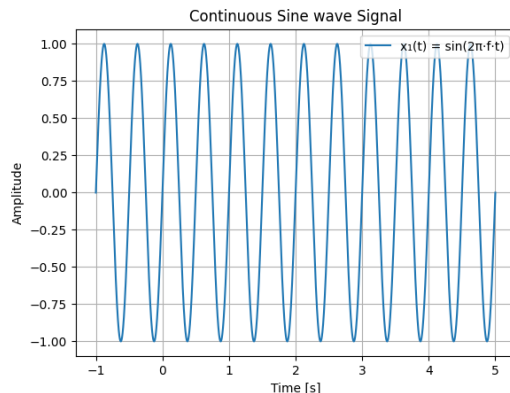
- Señal senoidal: $x_1(t) = \sin(2\pi \cdot f \cdot t)$
- Señal exponencial: $x_2(t) = e^{(-2t)} \cdot u(t)$
- Señal triangular: $x_3(t) = \text{tri}(t, f)$
- Señal cuadrada: $x_4(t) = \text{sq}(t, f)$

Proceso de graficación

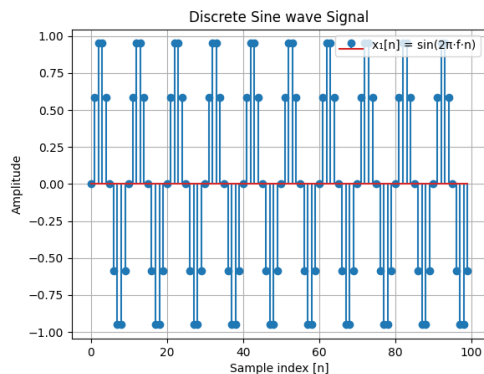
Para graficar las señales continuas, se utilizó la función `linspace` de NumPy con al menos 1000 puntos en el intervalo de tiempo $t \in [-1, 5]$ s (ajustado según el tipo de señal). Las señales discretas fueron obtenidas mediante muestreo con un periodo adecuado, por ejemplo, $T_s = 0.01$ s. Las funciones personalizadas `continuous_plotter` y `discrete_plotter` del módulo `grapher.py` fueron usadas para graficar cada señal con títulos, leyendas y etiquetas de ejes.

Gráficas generadas

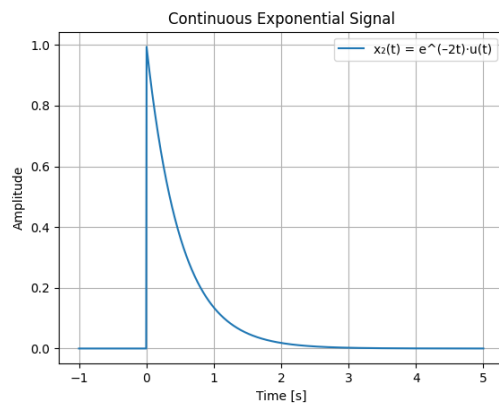
Señal Senoidal continua



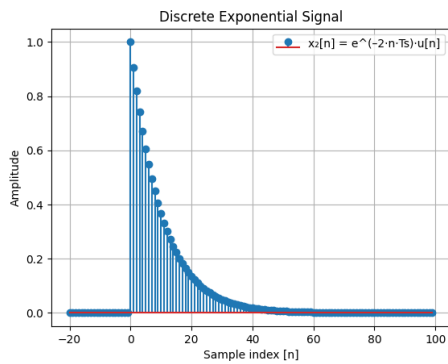
Señal Senoidal discreta



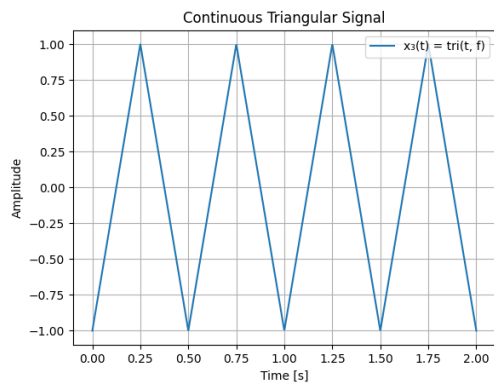
Señal Exponencial continua



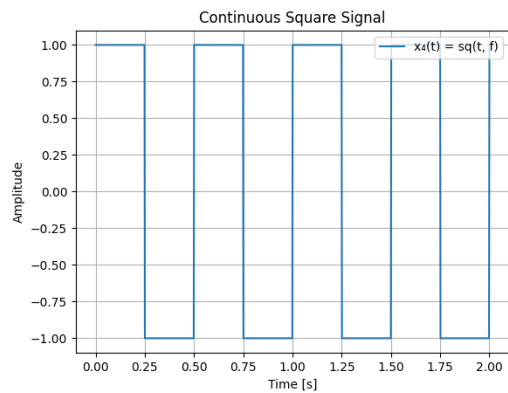
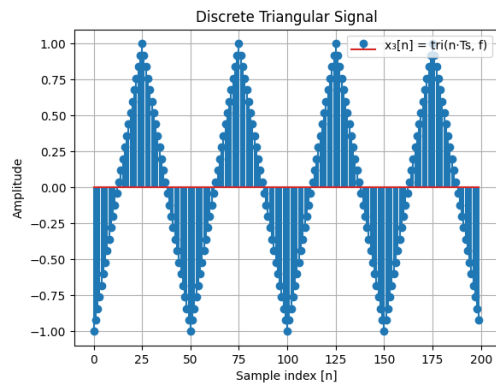
Señal Exponencial discreta



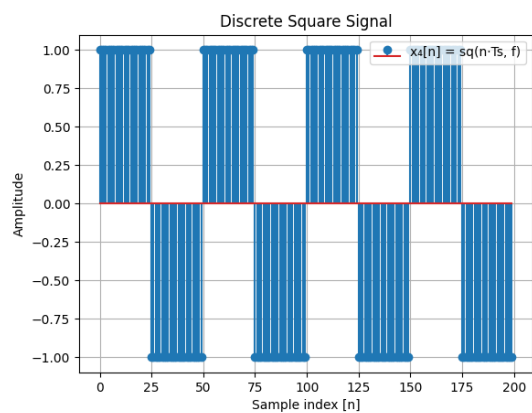
Señal Triangular continua



Señal Triangular discreta



Señal Cuadrada discreta



Librerías utilizadas

Se utilizaron las siguientes librerías de Python:

- NumPy: para generar vectores de tiempo y calcular valores de las señales.
- SciPy: para generar señales triangulares y cuadradas.
- Matplotlib: para graficar las señales.

Repositorio en GitHub

URL del repositorio: [Yan-carlo-1311/Pds](https://github.com/Yan-carlo-1311/Pds)