

**Instituto Politécnico Nacional**  
**Escuela superior de Cómputo**  
Algoritmos Bioinspirados  
Práctica No. 1: Introducción a Python  
Realizada por: Calva Guzmán Alan Alexis  
Profesor Jorge Luis Rosas Trigueros  
Realizada: 02/03/2025  
Fecha de entrega: 03/03/2025

**Marco teórico:**

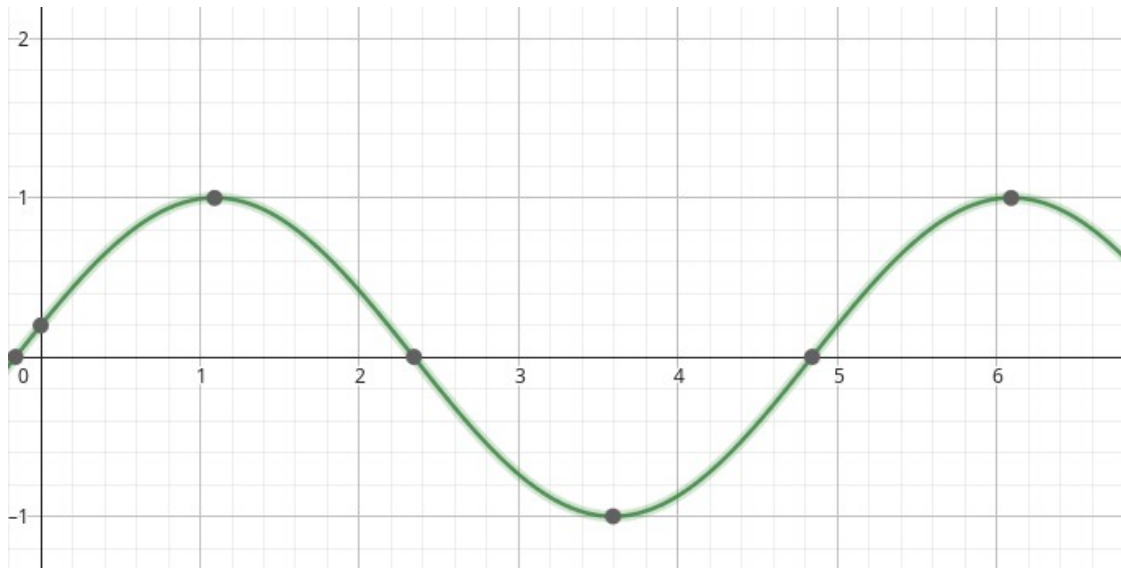
Esta práctica tiene como objetivo brindar un acercamiento a Python, un lenguaje de programación ampliamente aplicado a modelos contemporáneos de sistemas informáticos de diversa índole. Fueron solicitados, en congruencia con la unidad de aprendizaje, tres funciones codificadas en Python: Una que genera una lista con 100 números de tipo coma flotante en un rango del -1000 al 1000, una función que identifique el número más grande en esta lista así como el menor, una función que grafique dos periodos de una función senoidal, resaltando los puntos en los que la pendiente de la función tangente asociada a la función es igual o cercana a cero.

La realización de la práctica demanda conocimientos básicos de cálculo, de programación y del lenguaje Python. Fueron recomendados el uso de la librería Numpy, que brinda utilidades muy convenientes para esta práctica. Respecto al cálculo, es conveniente para comprender algo de la teoría detrás de esta práctica, pues fué recomendado usar la función secante para aproximar los puntos en los que la tangente es cercana a cero, dado que al trabajar sobre dos periodos de una función senoidal proveniente de una transformación de la función seno, la función tangente asociada no puede colindar una sola vez con la función al ser su pendiente igual a cero; de este modo, es requerido valerse de otros métodos para calcular los puntos máximos y mínimos de la función.

La función senoidal a graficar está dada por  $f(t) = \sin(2\pi t/5 + 0.2)$

Dado que la derivada de una función senoidal permite determinar sus máximos y mínimos, es importante identificar los puntos donde , ya que en estos puntos la función cambia de tendencia. En este caso, la derivada de la función dada es:

Para encontrar los máximos y mínimos de la función, se deben localizar los valores de en los que , lo que implica encontrar los ceros de la función coseno. Esto se puede resolver mediante aproximaciones numéricas o el uso directo de funciones de Numpy como `argrextrema` de `scipy.signal`.



*Figura 1: Gráfica en geogebra como referencia de la función dada*

La representación gráfica de la función junto con los puntos críticos permitirá visualizar de manera clara los máximos y mínimos, facilitando el análisis de la transformación aplicada a la función seno original. Además, esta práctica refuerza el uso de estructuras de datos en Python y el análisis matemático aplicado a la programación científica.

### **Material y equipo:**

Fue utilizado como auxiliar en la escritura de código el programa Codium, fueron generados entornos de python con Anaconda, utilizando la versión 3.9 de python; todo esto corriendo en Ubuntu 24.04.2 en un equipo de escritorio con 8 Gb de memoria RAM y un procesador Ryzen de 64 bits como características principales relevantes.

### **Desarrollo de la práctica:**

La práctica se desarrolló implementando un programa en Python que aborda los tres problemas planteados. Primero, se generó una lista de 100 números aleatorios de tipo flotante dentro del intervalo de -1000 a 1000, utilizando la función `np.random.uniform()`. Posteriormente, se recorrió la lista para identificar el número más grande y el más pequeño mediante una estructura iterativa `for`, comparando cada elemento con valores iniciales predefinidos.

Para la segunda parte de la práctica, se construyó la gráfica de la función senoidal en el intervalo de dos periodos de la función. Se generaron valores de utilizando `np.linspace()`, asegurando una resolución adecuada en la visualización.

La derivada de la función se calculó como  $\cos(x)$ , y se identificaron los puntos donde esta derivada se aproximaba a cero. Estos puntos corresponden a los valores críticos de la función, es decir, sus máximos y mínimos. Para resaltarlos en la gráfica, se utilizaron puntos de color rojo.

Finalmente, la representación visual se realizó mediante `matplotlib.pyplot`, incorporando etiquetas, una cuadrícula y una leyenda para mejorar la interpretación de los resultados. La gráfica obtenida muestra claramente la función senoidal y los puntos donde la pendiente es nula, confirmando los cálculos analíticos realizados previamente, y coincidiendo con la referencia obtenida en geogebra.

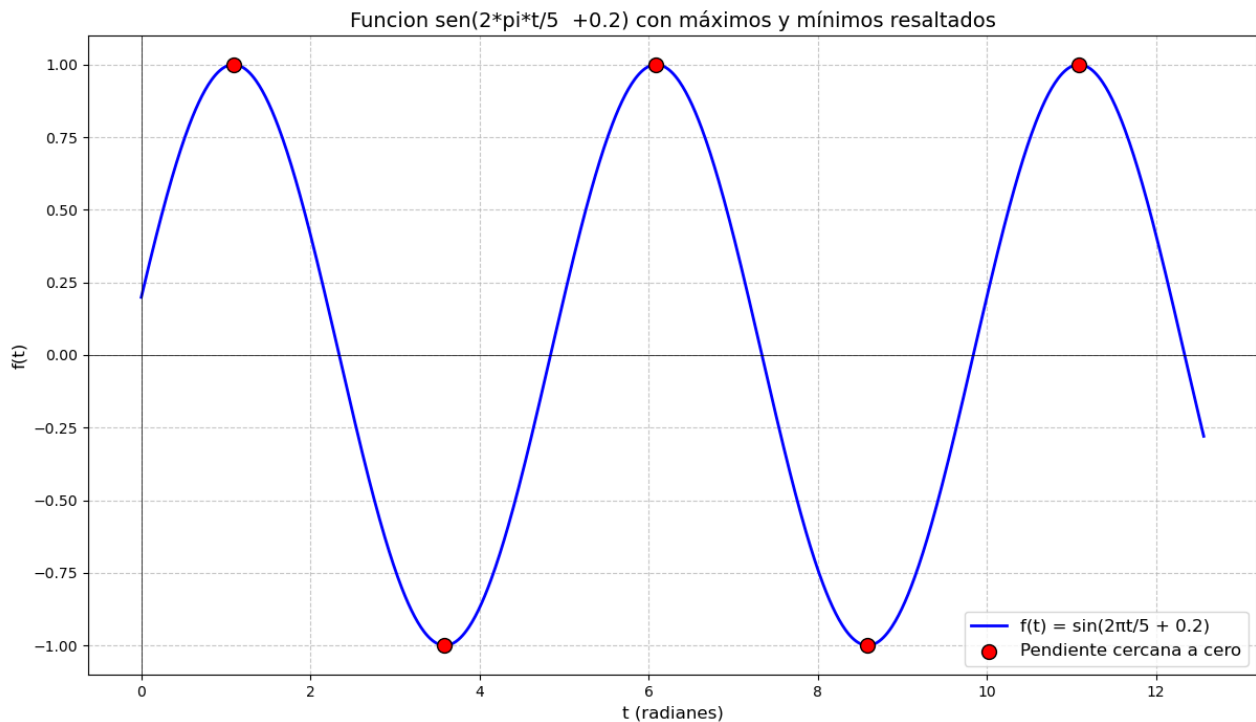


Figura 2: Gráfica generada por el código de Python

### **Conclusiones y recomendaciones:**

Para realizar una práctica de este estilo, es ampliamente recomendable consultar y utilizar (si no hay restricciones al respecto) las funciones de python que faciliten algunas tareas, como algunas que fueron usadas en la práctica y otras que fueron consideradas pero no utilizadas.

De la misma forma, sabiendo con qué se va a trabajar, refiriendome específicamente a las listas, resulta conveniente explorar las librerías (para este caso `numpy`, principalmente), con el objetivo de saber qué ofrecen que pueda resultar de utilidad al manejar elementos y funciones pertenecientes a estas.

Como siempre resulta para un trabajo de ingeniería aplicada, el fundamento matemático detrás de una implementación no sobra; y aunque en ocasiones es muy profundo como para intentar entenderlo en su totalidad, investigarlo como mínimo a grandes razgos plantea una ventaja no despreciable.