

Tarea No.1: Gráfica senoidal en Octave y Python

Marianne Nicté, Rodríguez Canek, 202000656^{1,*}

¹Facultad de Ingeniería, Departamento de Electronica,
Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.

Git es un sistema de control de versiones distribuido que permite gestionar los cambios realizados en los archivos de un proyecto. Su uso facilita el trabajo colaborativo entre varios programadores, ya que cada uno puede trabajar de manera independiente en una copia local del repositorio. A través de comandos como *commit*, *push* y *pull*, los desarrolladores sincronizan sus avances y mantienen actualizada la versión general del proyecto. Además, Git permite la creación de ramas para desarrollar nuevas funciones o corregir errores sin afectar el código principal. De esta manera, el control de versiones garantiza un trabajo coordinado, seguro y organizado dentro de los equipos de desarrollo.

I. OBJETIVOS

A. Generales

- Aprender a desarrollarse en diferentes lenguajes de programación

B. Específicos

- * Generar una grafica de una funcion senoidal en Octave
- * Generar una grafica de una funcion senoidal en Octave
- * Aprender acerca de los repositorios y sistema de control Git

MARCO TEÓRICO

Sistema de Control de Versiones Git

Git es un sistema de control de versiones distribuido, diseñado para registrar y gestionar los cambios realizados en archivos de un proyecto a lo largo del tiempo. Permite a los desarrolladores trabajar de manera colaborativa, sincronizando sus contribuciones sin sobrescribir el trabajo de otros. Git almacena los cambios en repositorios locales y permite sincronizarlos con servidores remotos, como GitHub o GitLab. Sus principales funciones incluyen la creación de versiones o *commits*, el seguimiento de la historia de cambios, la posibilidad de ramificar (*branching*) para trabajos paralelos y la integración de distintas ramas mediante fusiones (*merges*). Gracias a su diseño distribuido, cada copia del repositorio contiene todo el historial del proyecto, proporcionando seguridad y flexibilidad para el desarrollo de software.

II. RESULTADOS



Figura 1: Simbolo de Git

Python y Octave

Python es un lenguaje de programación de propósito general, conocido por su sintaxis clara y su gran versatilidad. Se utiliza en diversos campos, como desarrollo web, inteligencia artificial, ciencia de datos y automatización. En aplicaciones científicas y de ingeniería, se apoya en bibliotecas especializadas como NumPy, Matplotlib y SciPy para el cálculo numérico y la visualización de datos.



Figura 2: Simbolo de Git

Por otro lado, Octave es un lenguaje de programación orientado principalmente al cálculo numérico y al análisis matemático. Su sintaxis es muy similar a la de MATLAB, lo que lo convierte en una opción popular para tareas de simulación, álgebra lineal y modelado matemático. Aunque Python es más versátil en cuanto a aplicaciones generales, Octave destaca en entornos académicos y de ingeniería donde se requieren cálculos matemáticos complejos de forma rápida y directa.

* e-mail: 3243383091703@ingenieria.usac.edu.gt



Figura 3: Simbolo de Octave

En resumen, Python es un lenguaje multipropósito con un amplio ecosistema de librerías, mientras que Octave está más enfocado en el ámbito del cálculo numérico y científico, sirviendo como una alternativa gratuita a MATLAB.

III. MARCO PRÁCTICO

A. Octave

El programa en Octave genera un vector x desde 0 hasta 2π con incrementos de 0.01 y luego calcula el seno de cada valor para formar el vector y . Finalmente, grafica y en función de x , mostrando la curva de la función seno en ese intervalo.

B. Python

En este programa en Python se importa la librería NumPy para cálculos numéricos y Matplotlib para gráficos. Primero se genera un vector X con valores desde 0 hasta 2π con un incremento de 0.01. Luego se calcula $Y = \sin(X)$ y se grafica Y contra X . Después se limitan los ejes: el eje X de 0 a 2π y el eje Y de -1.5 a 1.5 para visualizar mejor la gráfica. Finalmente, se muestra la curva con `plt.show()`.

IV. CODIGO

```
GNU Octave, version 10.2.0
Copyright (C) 1993-2025 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

Home page:      https://octave.org
Support resources: https://octave.org/support
Improve Octave: https://octave.org/get-involved

For changes from previous versions, type 'news'.

octave:1> x = 0:0.01:2*pi;
octave:2> y = sin(x);
octave:3> plot(x,y);
octave:4>
```

Figura 3: Código en Octave.

```
seno.py X
C: > Users > 50241 > Desktop > Tareasgit > seno.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.arange(0, 2 * np.pi, 0.01) # Rango de 0 a
5 y = np.sin(x) # Calcular seno
6
7 plt.plot(x, y) # Graficar
8 plt.xlim(0, 2 * np.pi) # Limitar eje X
9 plt.ylim(-1.5, 1.5) # Limitar eje Y
10 plt.show() # Mostrar gráfica
```

Figura 4: Código en python.

V. RESULTADOS

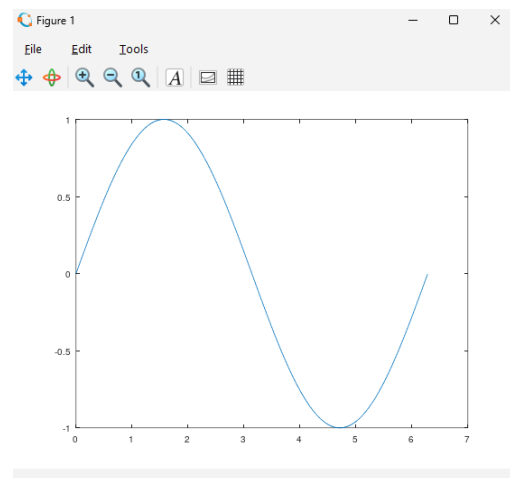


Figura 5: Gráfica de octave.

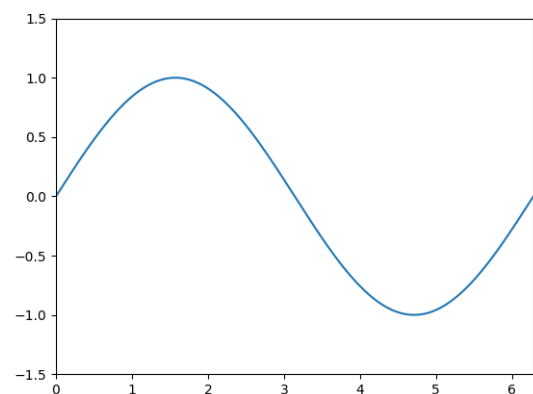


Figura 6: Gráfica de python.

VI. CONCLUSIONES

- El uso conjunto de Octave y Python demuestra cómo diferentes entornos de programación pueden ser

empleados para resolver un mismo problema matemático, fomentando la adaptabilidad del estudiante a diversas herramientas según el contexto o los recursos disponibles.

- La realización de esta tarea no solo permitió afianzar conocimientos en programación y matemáticas, sino que también promovió competencias clave en ingeniería como el análisis numérico, la representación gráfica de fenómenos y el uso de software

especializado.

- Aunque Git suele relacionarse con entornos profesionales, su incorporación en actividades académicas favorece la organización del código, el seguimiento de cambios y la adquisición temprana de buenas prácticas para el desarrollo colaborativo y estructurado de software.

-
- [1] Oppenheim, A. (2000). *Señales y sistemas*. Prentice Hall.
 [2] Reckdahl, K. (Versión [3.0.1]). (2006). *Using Imported Graphics in LATEX and pdfLATEX*.
 [3] Haley, S.(Feb. 1983).*The ThÄ©venin Circuit Theorem*

and Its Generalization to Linear Algebraic Systems. Education, IEEE Transactions on, vol.26, no.1, pp.34-36.