

NumPy - Temario A -

Esta hoja de trabajo tiene como objetivo reforzar el uso del módulo NumPy de Python y ver su importancia en la resolución de problemas.

Ejercicio #1 - Generación de gráficas con Matplotlib

El propósito de la computación científica no son los números en sí. Más bien, es tener el conocimiento, la intuición y la comprensión de los datos, lo que en inglés se llama *insight*. Para entender el significado de los muchos números que computamos, a menudo se necesita hacer un post-proceso, análisis estadístico y/o visualización gráfica de nuestros datos. En este ejercicio trabajaremos con los módulos de Python vistos en clase NumPy y Matplotlib. Esta librería permite generar gráficas de alta calidad del tipo $y = f(x)$ (y mucho más).

En un nuevo archivo de Thonny copie el siguiente código:

```
import numpy as np                # importar Numpy para trabajar con arreglos rápidos
import matplotlib.pyplot as plt   # importar las funciones para graficar

x = np.arange(-3.14, 3.14, 0.01)  # crear datos para x
y = np.sin(x)                    # calcular los valores de y

plt.plot(x, y)                   # crear la gráfica
plt.show()                       # mostrar la grafica
```

Guarde su archivo con el nombre `Ejercicio1_Numpy.py` y ejecútelo. Esta gráfica muestra los datos pero viola varias convenciones importantes de la práctica de la ingeniería. Los ejes no están rotulados con un nombre, no cuenta con una leyenda que describa la función graficada, no hay ningún tipo de cuadrícula, y tampoco tiene un título apropiado.

[15 pts] Complete el primer ejercicio agregando los elementos que faltan en la gráfica. (En el módulo de Numpy en Canvas y la sección de referencia de esta hoja de trabajo puede encontrar la documentación oficial de Matplotlib. También puede serle útil - <https://www.edureka.co/blog/python-matplotlib-tutorial/>)

Entregar en Canvas:

- **Ejercicio1_Numpy.py** código fuente del programa, debidamente comentado y con encabezado (Nombre, carné, fecha y descripción).

NumPy - Temario A -

Ejercicio #2 - Ingeniería: Fábrica de juguetes

Una fábrica de juguetes pequeños acaba de recibir un pedido de una compañía de comida rápida para que le fabriquen tres juguetes, de bajo costo, para incluirlos en las cajas de comida de niños. La fábrica tiene que producir juguetes para niños (juguete M), niñas (juguete F) y una versión genérica (juguete G). Además, basado en la demanda y estadísticas demográficas, la compañía de comida rápida ha especificado que deben producirse un 5% más de juguetes para niñas que para niños, y que no hay restricción sobre el número de juguetes genéricos a ser producidos. Los componentes de cada juguete (M, F y G) serán moldeados de plástico inyectado (Proceso 1) y luego ensamblados (Proceso 2). Luego de haber diseñado los juguetes, se determina que los siguientes tiempos de producción son los necesarios para cada juguete:

- Se requieren 2 minutos para moldear 6 juguetes (M), y un minuto para ensamblar los 6 juguetes
- Se requieren 2 minutos para moldear 12 juguetes (F), y 8 minutos para ensamblar los 12 juguetes
- Se requieren 4 minutos para moldear 6 juguetes (G), y 2 minutos para ensamblar los 6 juguetes

El ingeniero industrial de la fábrica se ha enfermado por lo que les han solicitado a los estudiantes del curso de Algoritmos y Programación básica, que determinen el número máximo de juguetes, de cada tipo, que se pueden producir en un día

Para resolver el problema se asume lo siguiente:

- La máquina de moldeo por inyección requiere de mantenimiento diario. Por este motivo, ésta solamente puede ser operada durante un máximo de 756 minutos de un total de 1440 minutos que hay en el día.
- La línea de ensamblaje trabaja en tres turnos diarios, incluyendo descansos, por un máximo de 1260 minutos de los 1440 minutos en el día.

[20 pts.] Realice el análisis del problema:

1. ¿Con qué va a trabajar?
 - a. ¿Qué información se tiene?
 - b. ¿Qué otros datos se necesitan?
2. ¿Qué condiciones o restricciones tiene el problema? Es decir, ¿qué debe tomar en cuenta para su diseño?
3. ¿Cuál debe ser el resultado final de su solución? Es decir, ¿qué debe poder ver el usuario al final de la ejecución?

[20 pts.] Realice el diseño de la solución:

1. Elabore un diagrama de flujo o pseudocódigo para mostrar la solución.

Entregar en Canvas: **Ejercicio2_Numpy.pdf**: archivo PDF con el diagrama de flujo del programa

NumPy - Temario A -

Ejercicio #3 - Del diseño a la práctica

[35 pts] Utilizando el diseño que hizo en el ejercicio 2, escriba un programa en Python. Recuerde que:

1. [05 pts.] El código debe ir debidamente comentado.
2. [05 pts.] El programa debe utilizar arreglos y/o vectores NumPy.
3. [05 pts.] El programa genera la gráfica requerida, debidamente rotulada.
4. [10 pts.] El programa cumple con los requerimientos solicitados.
5. [10 pts.] El resultado es fácil de leer y es correcto.

Entregar en Canvas: **Ejercicio3_Numpy.py**: código fuente del programa.

Referencias:

- Módulo NumPy: <https://docs.python.org/3/library/stdtypes.html#string-methods>
- Módulo Matplotlib: https://matplotlib.org/users/pyplot_tutorial.html

Reflexión individual

[10 pts.] Al finalizar los ejercicios de esta hoja de trabajo, ingrese al sitio Canvas de su curso y complete las preguntas que aparecen en la Reflexión correspondiente a esta hoja.