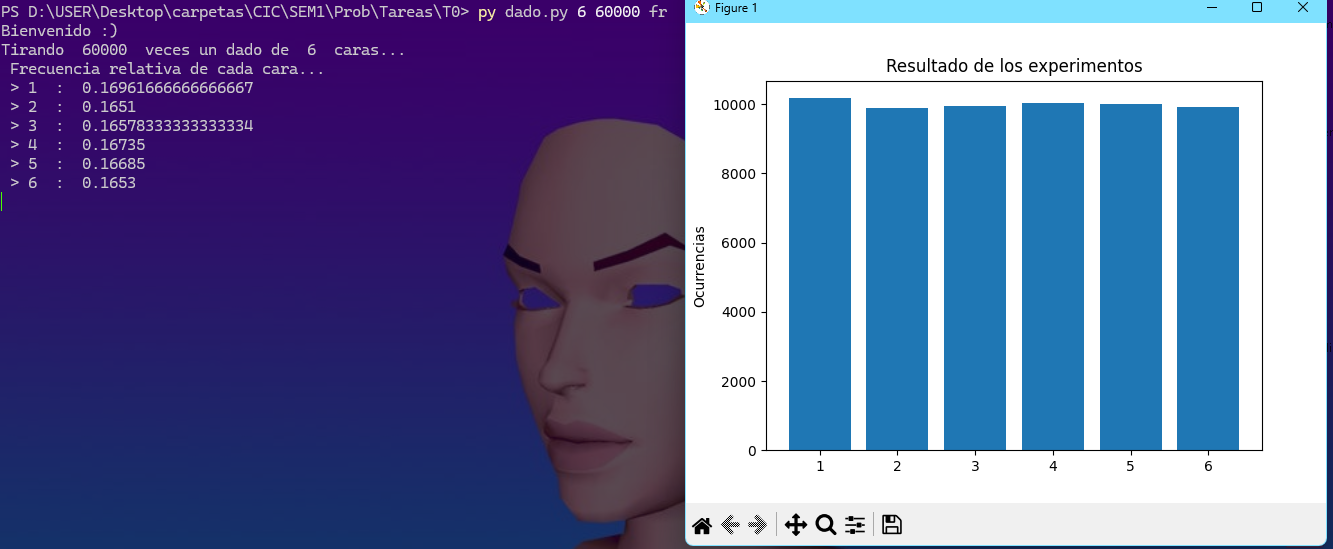
## Alumno: Martínez Alfaro Felipe de Jesús.

## Profesor: Juan Carlos Chimal Eguía.

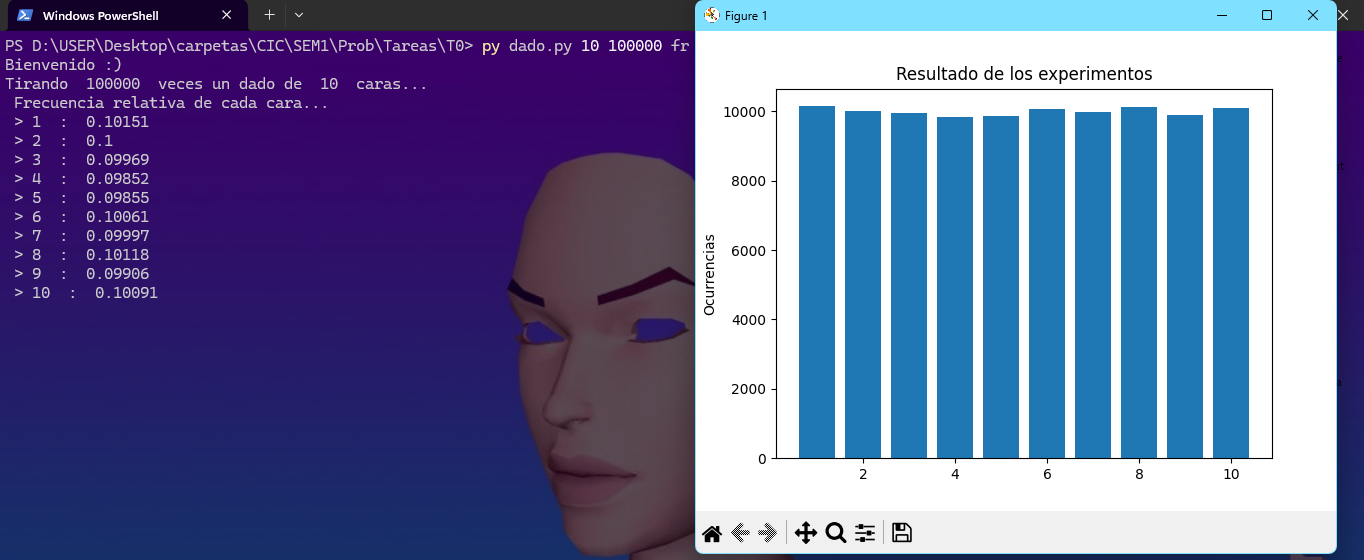
# Tarea 0 Dados.

Capturas de pantalla:

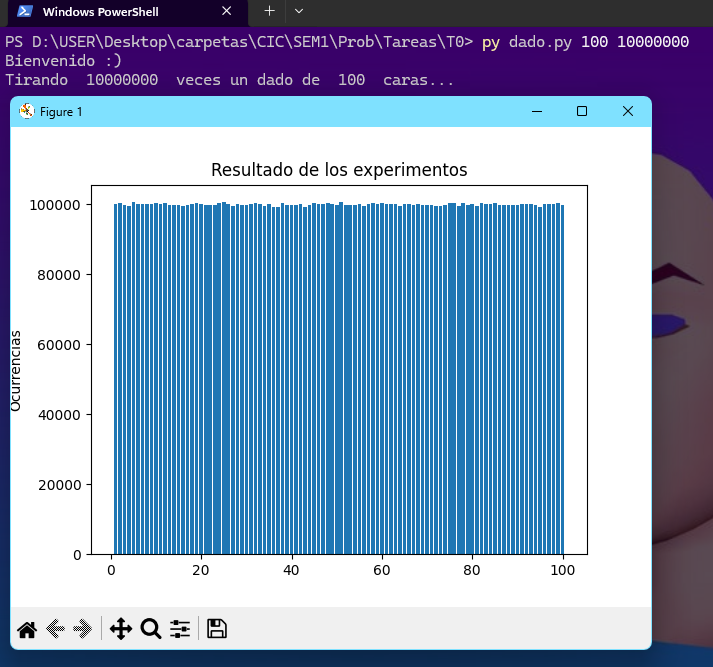
* Dado de 6 caras y 60,000 muestras, con sus frecuencias relativas.



Dado de 10 caras con 100,000 muestras, con sus frecuencias relativas.



Dado de 100 caras con 10,000,000 de muestras, sin frecuencia relativa.



Código del programa:

"""

Programador: Felipe de Jesus Martinez Alfaro.

Clase: Probabilidad, procesos aleatorios y estadistica.

Llamar el programa de la siguiente manera, dentro de la misma carpeta que el programa...

> py dado.py [caras] [muestras]

Esta clase simula la tirada de un dado indicandole el numero de caras y un numero de tiradas a contar.

Niveles de comentarios.

# 0 Documenta la linea o lineas siguientes...

# 1 Documenta la misma linea.

# 2 Documenta la misma linea.

# 3 Documenta la misma linea.

# 4 Documenta la misma linea.

# 5 Documenta la misma linea.

"""

""" Bibliotecas... """

**import** random; # Para el resultado aleatorio.

**import** sys; # Para el main.

**import** matplotlib.pyplot **as** plt; # Para graficar.

""" Funciones... """

**def** **tirarDados**(caras:int,tam\_muestra:int):

# Por si ponen algo raro...

**if** caras < 1 **or** tam\_muestra < 1: # Parametros fuera del dominio.

print("Pues entonces nada >:c",type(caras),"-",tam\_muestra)

**return** [];

# Se hacen las tiradas...

eventos : list = [0 **for** i **in** range(1,caras+1)] # Tabla de ocurrencias.

**for** i **in** range(tam\_muestra):

eventos[random.randrange(1,caras+1) - 1] += 1; # Seleccion aleatoria y agrega la ocurrencia.

**return** eventos;

**def** **histograma**(caras:int, muestras:list):

#Configuro lo necesario para graficar...

fig, ax = plt.subplots() # Inicializo grafica.

labels = range(1,caras+1) # Pongo las caras en el eje x.

ax.bar(labels, muestras) # Agrego la tabla de ocurrencias.

#Nombro ejes y mando a graficar...

ax.set\_ylabel('Ocurrencias')

ax.set\_title('Resultado de los experimentos')

plt.show()

**pass**;

""" MAIN!!! """

**def** **main**(params:list = sys.argv)->int:

# Variables a usar..."""

dado:str # Numero de caras.

n\_exp:str # Tam. del muestreo.

ans:list # Resultados de los eventos.

print("Bienvenido :) ")

# Obtenemos los parametros...

**if**( len(sys.argv) < 3 ): # Por si no hay parametros al llamar la funcion.

dado = input("Caras: ")

n\_exp = input("Muestras: ")

**else**: # Se obtienen los parametros si los hay.

dado , n\_exp = params[1] , params[2]

**if**(**not** dado.isdigit() **or** **not** n\_exp.isdigit()): # Verificamos que se ingresacen numeros.

print("Parametros incorrectos")

**return**;

print("Tirando ",n\_exp," veces un dado de ",dado," caras...")

# Se llaman las funciones...

ans = tirarDados( int(dado) , int(n\_exp) ) # Muestreo.

# Indico la frecuencia relativa...

**if** len(params) > 3 :

print(" Frecuencia relativa de cada cara...")

**for** i **in** range(len(ans)):

print(" >",i+1," : ",ans[i]/int(n\_exp));

# Imprimo el histograma

histograma(int(dado), ans) # Grafica.

**if** \_\_name\_\_ == '\_\_main\_\_':

main();