

Alumno: Martínez Alfaro Felipe de Jesús.

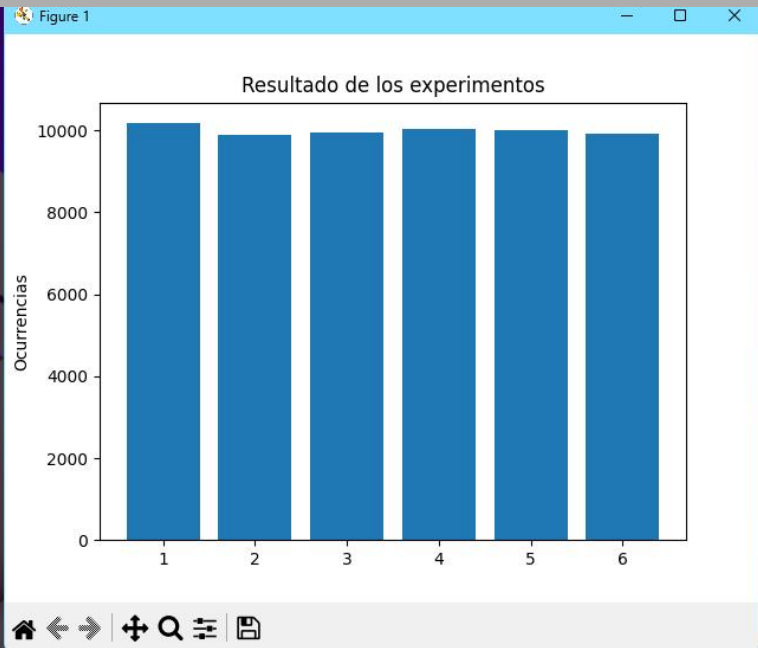
Profesor: Juan Carlos Chimal Eguía.

## Tarea 0 Dados.

### Capturas de pantalla:

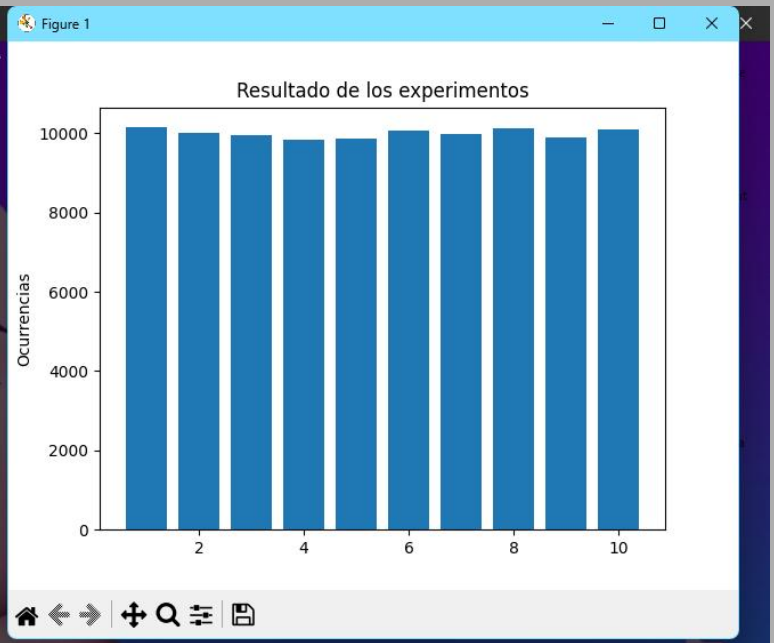
- Dado de 6 caras y 60,000 muestras, con sus frecuencias relativas.

```
PS D:\USER\Desktop\carpetas\CIC\SEM1\Prob\Tareas\T0> py dado.py 6 60000 fr
Bienvenido :)
Tirando 60000 veces un dado de 6 caras...
Frecuencia relativa de cada cara...
> 1 : 0.16961666666666667
> 2 : 0.1651
> 3 : 0.16578333333333334
> 4 : 0.16735
> 5 : 0.16685
> 6 : 0.1653
```

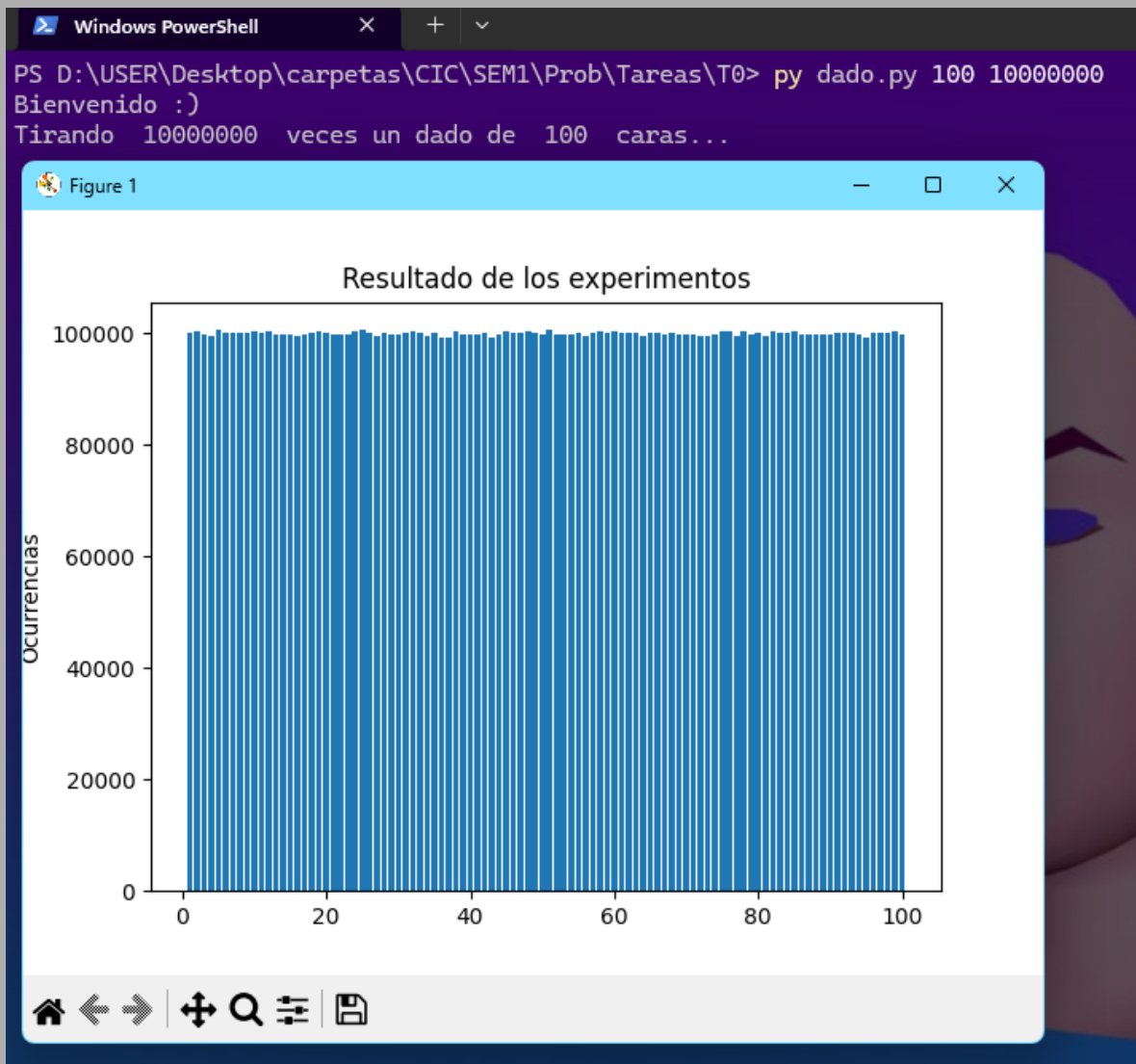


Dado de 10 caras con 100,000 muestras, con sus frecuencias relativas.

```
Windows PowerShell
PS D:\USER\Desktop\carpetas\CIC\SEM1\Prob\Tareas\T0> py dado.py 10 100000 fr
Bienvenido :)
Tirando 100000 veces un dado de 10 caras...
Frecuencia relativa de cada cara...
> 1 : 0.10151
> 2 : 0.1
> 3 : 0.09969
> 4 : 0.09852
> 5 : 0.09855
> 6 : 0.10061
> 7 : 0.09997
> 8 : 0.10118
> 9 : 0.09906
> 10 : 0.10091
```



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 estadística.

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 indicándole el número de caras y un número de tiradas a
contar.

Niveles de comentarios.
# 0 Documenta la línea o líneas siguientes...
# 1 Documenta la misma línea.
# 2 Documenta la misma línea.
# 3 Documenta la misma línea.
# 4 Documenta la misma línea.
# 5 Documenta la misma línea.

"""

""" 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();

```