
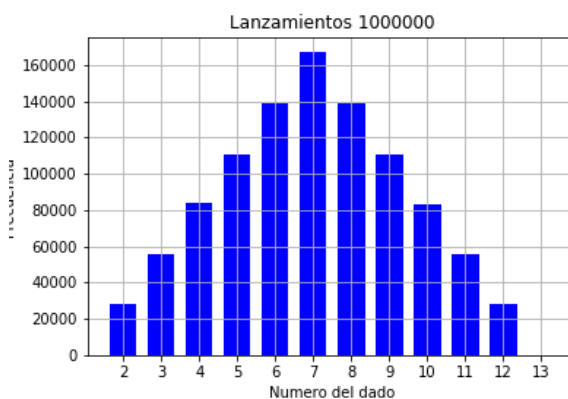
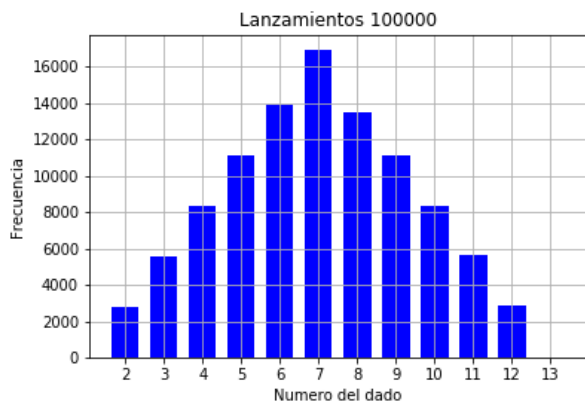
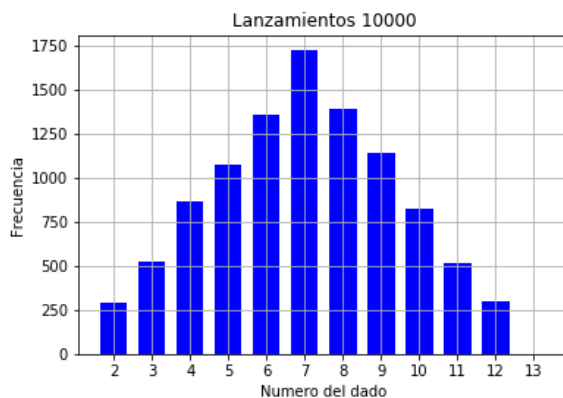
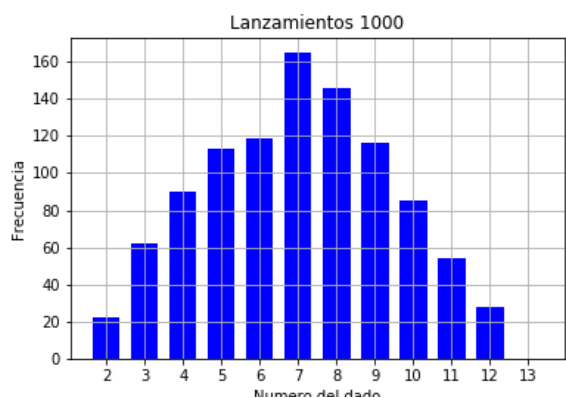
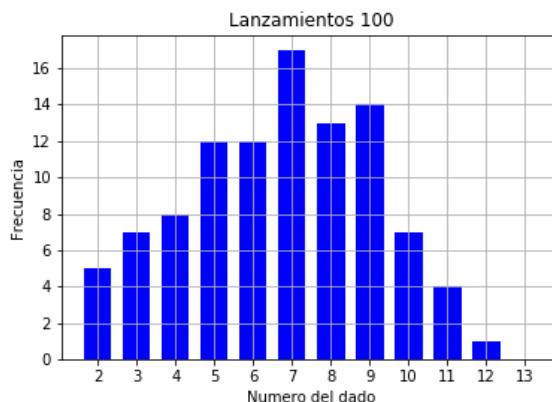
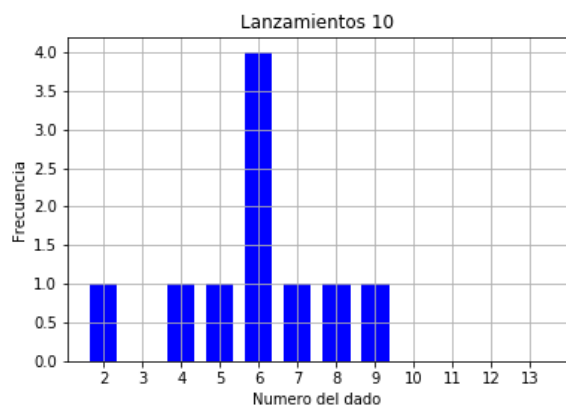

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2021/04/17
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		PRÁCTICA DE LABORATORIO	
CARRERA: Ingeniería en Computación		ASIGNATURA: Simulación	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Histograma de frecuencia de los datos	
ACTIVIDADES DESARROLLADAS			
<p>1. Desarrollar una aplicación en cualquier lenguaje que genere un histograma con el numero de frecuencia de la sumatoria, teniendo diferentes escenarios de lanzamientos:</p> <p>-10 -100 -1000 -10000 -100000 -1000000</p>			
<pre> """ Created on Sat Apr 17 17:57:14 2021  @author: EstebanRM """  import matplotlib.pyplot as plt import numpy as np import random  def generadorHistograma(size):     print("Lista"+str(size))     list = [random.randint(1,6) + random.randint(1,6) for i in range(size)]     fig, hs = plt.subplots()     hs.grid(True)     hs.set_xticks(range(2,15))     hs.set_title("Lanzamientos " + str(size))     hs.set_xlabel("Numero del dado")     hs.set_ylabel("Frecuencia")     hs.hist(list, bins=np.arange(2,15)-0.5, rwidth=0.7, facecolor='blue')     fig.savefig('histograma/lanzamiento'+str(size)+'.png')  def jugar():     print("jugando")     dados = [10, 100, 1000, 10000, 100000, 1000000]     for cantidad in dados:         generadorHistograma(cantidad)  if __name__ == "__main__":     print("Ingreso a jugar")     jugar() </pre>			

## 2. Histogramas



En la practica del lanzamiento de 2 dados, como podemos observar en los histogramas mientras mayor era el numero de lanzamientos la frecuencia de resultado recaia en el numero 7, obteniendo como resultado que mientras mayor numero de lanzamientos mas probable sea que se obtenga el numero 7 como respuesta.

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2021/04/17
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

### CONCLUSIONES:

- Al crear la aplicación se pudo observar que mientras menor era el número de lanzamientos, la obtención de resultados variaba a una probabilidad de  $1/6$  por cada dado, pero cuando mayor era el número de lanzamientos el porcentaje de probabilidad de la suma tendía a la suma de 7.  
El cual podemos observar que mientras mayor sea el número de iteraciones la respuesta a la suma de un valor pseudoaleatorio es 7.

### RECOMENDACIONES:

Para crear la aplicación de suma con valores aleatorios es recomendable el uso del lenguaje de programación Python.

**Nombre de estudiantes:** Esteban Rosero



**Firma de estudiantes:**