

## Parte 3 - Lab3 Modelacion y Simulacion

- Marco Jurado 20308
- Cristian Aguirre 20231
- Diego Cordova 20212

### Task 1

```
In [ ]: import numpy as np
        from scipy.stats import geom
        import matplotlib.pyplot as plt

        p = 0.3

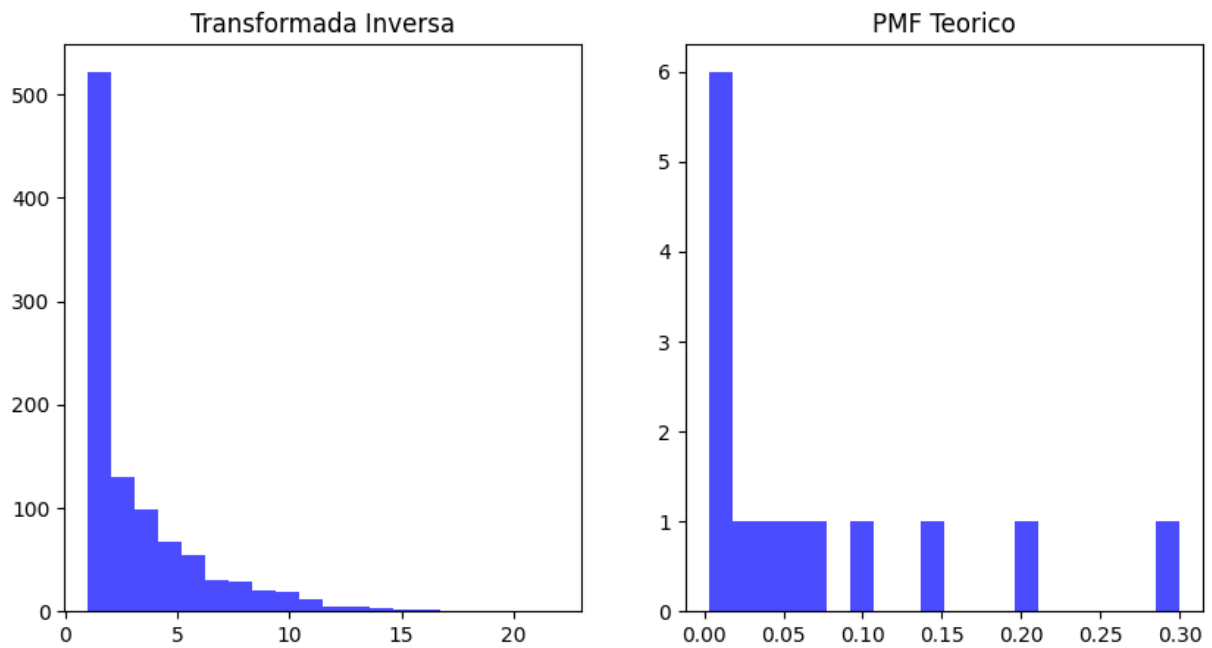
        # Transformada inversa
        U = np.random.uniform(0, 1, size=1000)
        sample = [int(np.ceil(np.log(1 - u) / np.log(1 - p))) for u in U]

        # PMF Teorico
        values = np.arange(1, 15)
        pmf_teorico = geom.pmf(values, p)

        # Trazo de histogramas
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

        ax1.hist(sample, bins=20, alpha=0.7, color='blue')
        ax1.set_title('Transformada Inversa')

        ax2.hist(pmf_teorico, bins=20, alpha=0.7, color='blue')
        ax2.set_title('PMF Teorico')
        plt.show()
```



## Task2

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

# Función de La distribución de Poisson con Lambda = 3
def poisson_pmf(k):
    return np.exp(-3) * (3 ** k) / np.math.factorial(k)

# Función de La distribución exponencial con beta = 1
def exponential_pdf(x):
    return np.exp(-x)

# Metodo del rechazo
sample = []
C = poisson_pmf(3) / (3 * np.exp(-3) / 2)

while len(sample) < 1000:
    x = np.random.exponential(scale=1)
    u = np.random.uniform()
    z = poisson_pmf(int(x)) / (C * exponential_pdf(x))

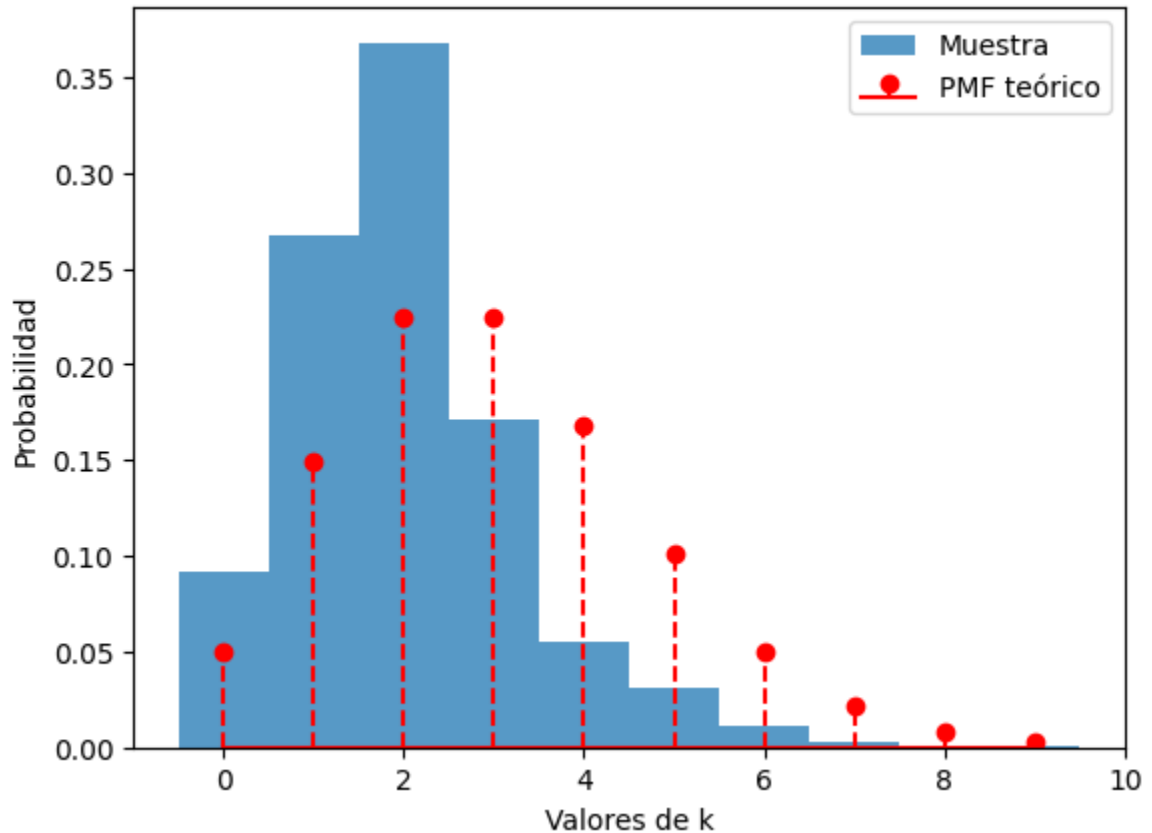
    if u <= z:
        sample.append(int(x))

# Graficar el histograma de la muestra generada
plt.hist(sample, bins=np.arange(-0.5, max(sample) + 1.5, 1), density=True, alpha=0.5)
k_values = np.arange(0, max(sample) + 1)
pmf_values = [poisson_pmf(k) for k in k_values]
plt.stem(k_values, pmf_values, linefmt='r--', markerfmt='ro', basefmt='r', label="PMF Teorico")
plt.xlabel("Valores de k")
plt.ylabel("Probabilidad")
plt.legend()
plt.title("Histograma de la muestra generada y PMF teórico de Poisson (λ=3)")
```

```
plt.show()
```

C:\Users\Diego-CB\AppData\Local\Temp\ipykernel\_17500\99643920.py:6: DeprecationWarning: `np.math` is a deprecated alias for the standard library `math` module (Deprecated Numpy 1.25). Replace usages of `np.math` with `math`  
 return np.exp(-3) \* (3 \*\* k) / np.math.factorial(k)

### Histograma de la muestra generada y PMF teórico de Poisson ( $\lambda=3$ )



### Tasks 3

```
In [ ]: import random

funcion = lambda x: (x**2) / 55

# Función de propuesta: Distribución uniforme discreta en el rango de 1 a 10
E = lambda: np.random.exponential(scale=10)

# Cálculo de la constante C
C = max(funcion(x) / (1/10) for x in range(1, 11))

# Método de aceptación-rechazo
def aceptacion_rechazo(size):
    samples = []
    while len(samples) < size:
        x = E()
        u = random.random()
        z = funcion(x) / (C * (1/10))

        if u <= z:
```

```
        samples.append(x)

    return samples

random_sample = aceptacion_rechazo(1000)

# Histograma de la muestra generada
import matplotlib.pyplot as plt

plt.hist(random_sample, bins=range(1, 12), density=True, align='left', rwidth=0.8,
plt.plot([funcion(x) for x in range(1, 11)], 'ro-', label='PMF objetivo')
plt.xticks(range(1, 11))
plt.xlabel('Valores')
plt.ylabel('Probabilidad')
plt.legend()
plt.show()
```

