

# Reporte Tarea 6

Victor Alejandro Oviedo Martínez

10 de noviembre de 2020

---

## 1. Introducccón

Para esta octava tarea[2] se ha estudiado el tema modelo de urnas, el cual es aplicado a la simulación de un sistema el cual tiene la función de coalescencia y fragmentación. En esencia, esta simulación tendrá una cantidad total de  $n$  partículas, con las cuales formaran un tamaño inicial de  $k$  cúmulos. Una vez teniendo los cúmulos se llevara acabo el proceso de fragmentación y coalescencia, en los cuales se dividirá los cúmulos para formar nuevos tamaños de los mismos, y luego volverán a unirse en el proceso de coalescencia. Este proceso podrá ser repetido un valor determinado.

Una vez determinado el valor de la variable  $n$ , generamos  $k$ , los cuales tendrán una distribución normal estándar para luego normalizarlos y convertirlos en enteros positivos que sumen a  $n$ . Para el proceso de fragmentación se utilizar la función sigmoideal centrada a la variable  $c$  la cual es el promedio de los cúmulos iniciales, esto con el fin de fragmentar con mayor probabilidad los cúmulos de mayor tamaño y inversamente a los cúmulos de menor tamaño. En el caso de coalescencia, se utiliza la función exponencial con el fin de agrupar con mayor probabilidad los cúmulos de menor tamaño, y con menor probabilidad los cúmulos de mayor tamaño. Se tiene que en estos dos procesos es donde se utiliza el modelo urnas, ya que al momento de iniciar estos procesos es necesario saber cuántos cúmulos tenemos y la cantidad de partículas que contienen cada uno de ellos. Por lo tanto, se tendrán diferentes ‘urnas’ para cada diferente cúmulos que contengan la misma cantidad de partículas. De esta forma se podrá tener el control de los diferentes cúmulos y el movimiento de los mismos. De esta forma se lleva acabo la simulación hasta un número determinado de iteraciones.

## 2. Desarrollo

Para esta séptima tarea se ha planteado el siguiente problema: Supongamos que cúmulos con  $c$  o más partículas (haciendo referencia al tamaño crítico  $c$ ) son suficientemente grandes para filtrar. Determina para diversas combinaciones de  $k$ ,  $n$  y el número de iteraciones  $t$  cuál porcentaje de las partículas se logra filtrar si el filtrado se lleva a cabo después de  $t$  iteraciones del proceso.

Para el desarrollo de esta tarea se ha utilizado el código ejemplo proporcionado por Schaeffer [1], el cual tiene el propósito de: simular el proceso de coalescencia y fragmentación con valores específicos  $n$  y  $k$ . Este código será tomado como base para entregar las características de esta tarea.

Para el desarrollo de esta tarea, se ha iniciado volviendo función el código proporcionado por Schaeffer [1], y agregando el código necesario para devolver como resultado de esta función el porcentaje de las partículas filtradas. Este procedimiento se lleva acabo al final del procedimiento de iteraciones y se eliminan todos los valores superiores a la variable  $c$ .

```

if paso == duracion-1:

    total = len(cumulos)
    filtrados = []
    nofiltrados = []
    for x in range(len(cumulos)):
        if cumulos[x] > c:
            filtrados.append(cumulos[x])
        else:
            nofiltrados.append(cumulos[x])

    PORfiltrados = ((len(filtrados)*100)/total)
    cumulos = nofiltrados

return PORfiltrados

```

Ya que el código regresa el valor requerido, se utilizara esta función para graficar los valores requeridos.

```

k = [200, 400, 600, 800]
n = [20000,40000,60000,80000]
duracion = [25, 50, 75, 100]
rep = 10

digitos = floor(log(len(k), 10)) + 1

DIM_1 = []
DIM_2 = []
DIM_3 = []
for K in k:
    DIM_3 = []
    for N in n:
        print(N)
        DIM_2 = []
        for DURACION in duracion:
            DIM_1 = []
            for i in range(rep):
                res = fil(K, N, DURACION)
                DIM_1.append(res)
            DIM_2.append(DIM_1)
        DIM_3.append(DIM_2)

```

Cómo se pudo ver en el código anterior, este será el encargado de obtener los valores para los diferentes valores de **k**, **n**, y **duracion**. Por último, se grafican los datos obtenidos.

### 3. Conclusión

Siguiendo las especificaciones de esta tarea se tiene como resultado la figura 1, 2, 3, y 4, en la cuales podremos encontrar para cada valor de **k**, **n**, y **duracion** su respuesta al porcentaje de cúmulos filtrados. Tomando en cuenta; **k** = [200, 400, 600, 800], **n** = [20000,40000,60000,80000], y **duracion** = [25, 50, 75, 100], esto con 10 repeticiones por combinación.

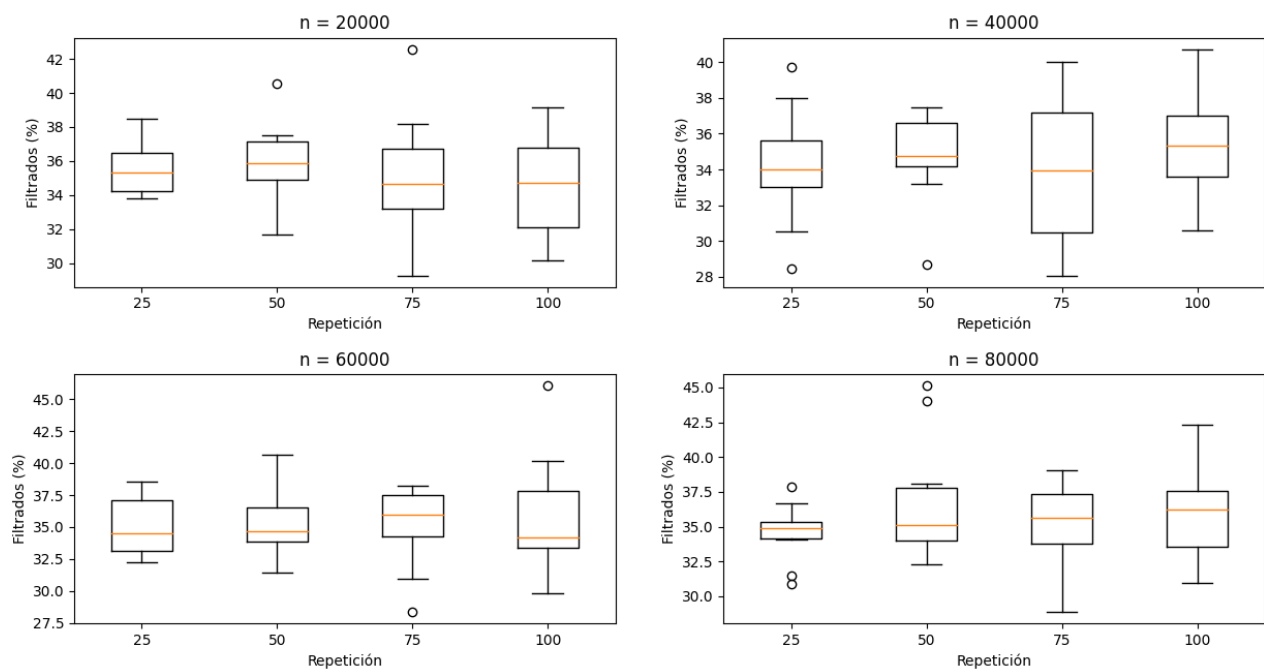


Figura 1:  $k = 200$ .

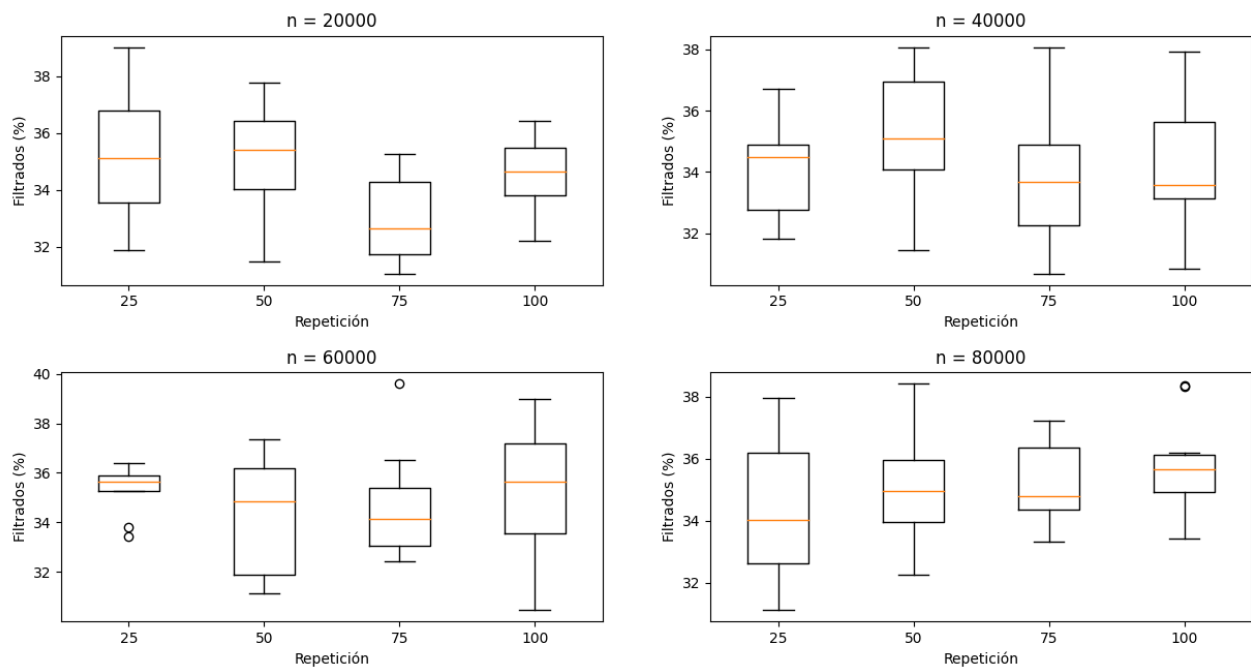


Figura 2:  $k = 400$ .

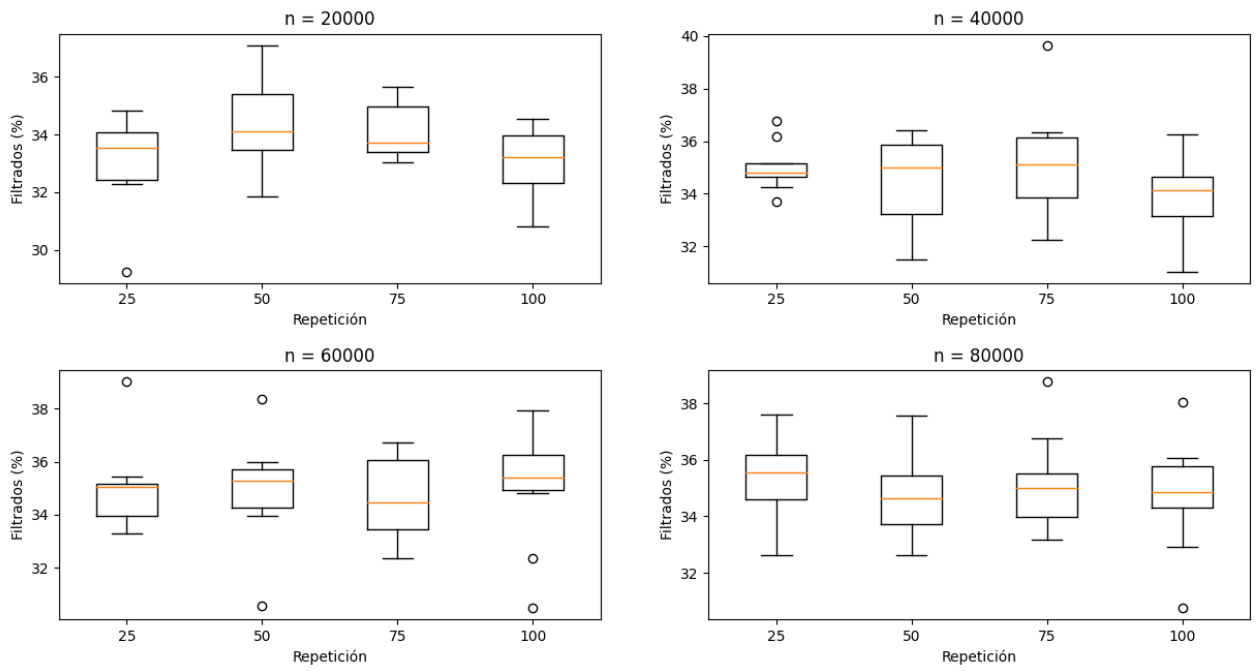


Figura 3:  $k = 600$ .

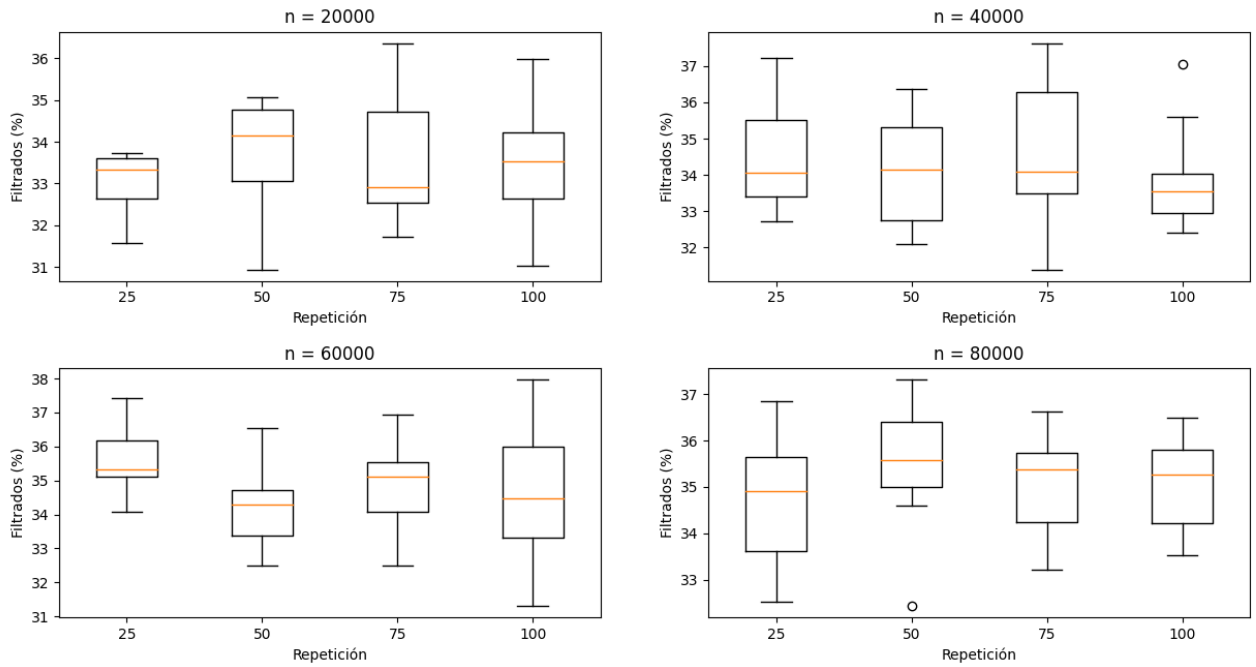


Figura 4:  $k = 800$ .

## Referencias

- [1] Elisa Schaeffer. `aggrfrag.py`, Nov 2020. <https://github.com/satuelisa/Simulation/blob/master/UrnModel/aggrFrag.py>.
- [2] Elisa Schaeffer. Práctica 8: modelo de urnas, Nov 2020. <https://elisa.dyndns-web.com/teaching/comp/par/p8.html>.