

# Práctica 8 modelo de urnas

Denisse Leyva

Abril 21, 2021

## 1. Introducción

La octava práctica es sobre fenómenos de coalescencia y fragmentación, donde partículas se unen para formar cúmulos y estos cúmulos se pueden volver a descomponer en fragmentos menores. Esto es relevante en muchos campos de química, como por ejemplo en el filtrado de aguas residuales, donde solamente los cúmulos de suficiente tamaño serán capturadas por el filtro y hay que buscar formas para facilitar que crezcan los cúmulos de residuos para lograr su filtrado adecuado.

Vamos a suponer que tenemos una cantidad total de  $n$  partículas y que al inicio el tamaño de los  $k$  cúmulos existentes sigue la distribución normal. Para lograr esto, vamos a crear  $k$  valores de la distribución normal estándar (media cero, desviación estándar uno) y luego normalizarlos para convertirlos en enteros positivos que sumen a  $n$  [2].

## 2. Objetivo

Supongamos que cúmulos con  $c$  o más partículas (haciendo referencia al tamaño crítico  $c$ ) son suficientemente grandes para filtrar. Gráfica  $k = 1000$ ,  $n \in 16k, 32k, 64k, 128k$  en cada iteración  $t$  el porcentaje de las partículas que se logra filtrar [2].

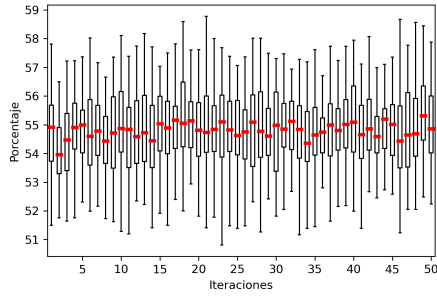
## 3. Código

```
1 def filtro(cumulos, c, n):
2     cum = []
3     for valor in cumulos:
4         if valor >= c:
5             cum.append(valor)
6     porcentaje = (sum(cum)*100)/n
7     return porcentaje
```

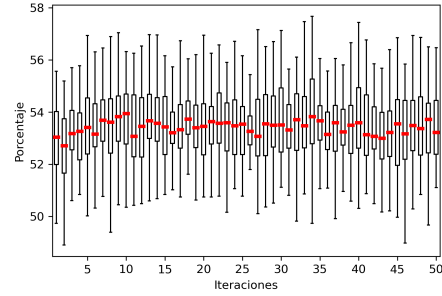
Código 1: Obtener el porcentaje del valor filtrado.

## 4. Resultados

Para obtener las gráficas se realizaron 50 réplicas con una duración de 50 pasos, por lo tanto nos muestran el porcentaje filtrado mayor o igual al valor crítico. El código completo se encuentra en GitHub [1]. El código base se obtuvo de Schaeffer [3].

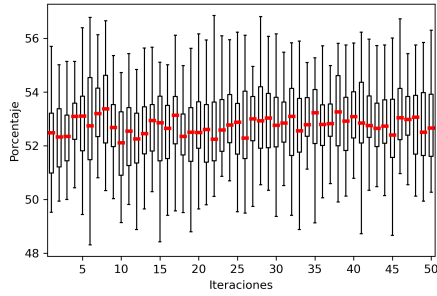


(a) 16000 partículas.

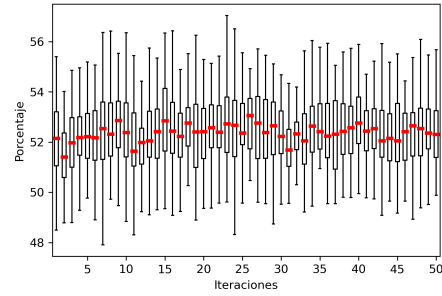


(b) 32000 partículas.

Figura 1: Porcentaje filtrado por iteración.



(a) 64000 partículas.



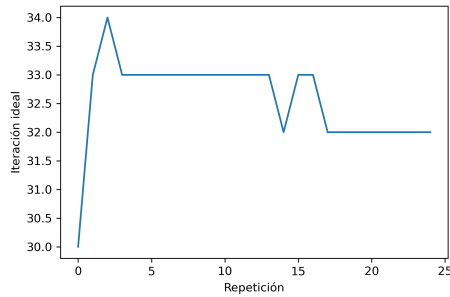
(b) 128000 partículas.

Figura 2: Porcentaje filtrado por iteración.

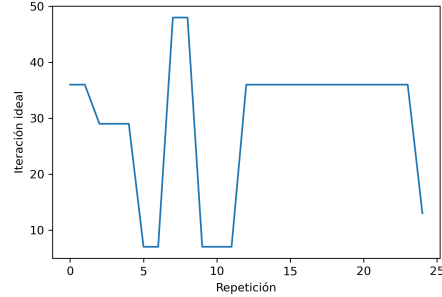
## 5. Reto 1

Como primer reto, determina si hay algún intervalo de iteraciones en el que el filtrado alcance un óptimo. Realiza réplicas para determinar si el momento en el cual se alcanza el máximo tiene un comportamiento sistemático. Incluye visualizaciones para justificar las conclusiones [2].

Para obtener las siguientes gráficas de las 50 réplicas con una duración de 50 pasps se repitieron 25 veces para de ahí sacar la mejor mediana por repetición. En el cuadro 1 se muestra el porcentaje de la iteración que más se repite. El código completo se encuentra en GitHub [1].



(a) 16000 partículas.



(b) 32000 partículas.

Figura 3: Gráfica de iteración ideal por repetición.

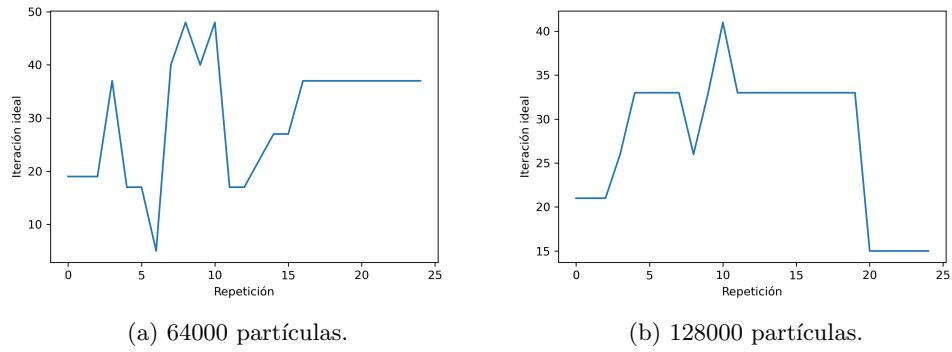


Figura 4: Gráfica de iteración ideal por repetición.

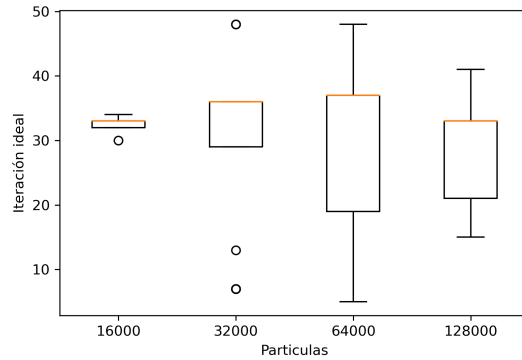


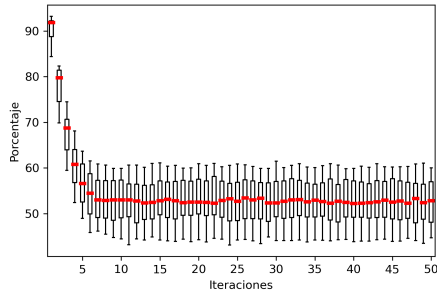
Figura 5: Gráfica caja bigote de iteración ideal.

Cuadro 1: Porcentaje de la mejor iteración para filtrar.

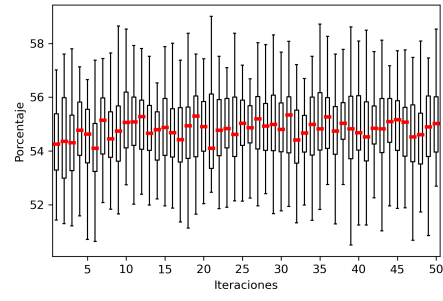
Partículas	Valor Ideal	Porcentaje
16000	33	56 %
32000	36	56 %
64000	37	40 %
128000	33	56 %

## 6. Reto 2

Como un segundo reto, determina cómo los resultados de la tarea y del primer reto dependen del valor de  $c$ . ¿Qué todo cambia y cómo si  $c$  ya no se asigna como la mediana inicial sino a un valor menor o mayor? [2].



(a) 16000 partículas en c1.



(b) 16000 partículas en c2.

Figura 6: Gráficas de porcentaje en diferentes puntos críticos.

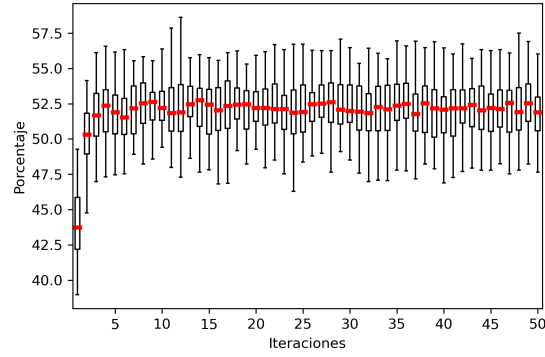
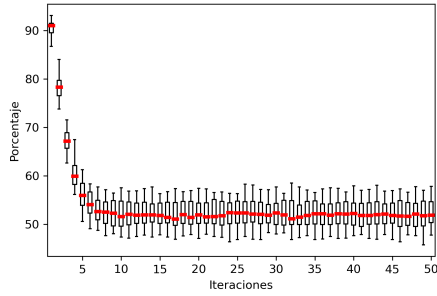
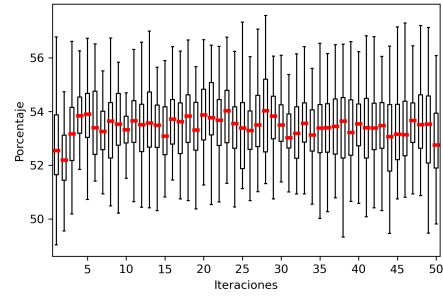


Figura 7: 16000 partículas en c3.



(a) 32000 partículas en c1.



(b) 32000 partículas en c2.

Figura 8: Gráficas de porcentaje en diferentes puntos críticos.

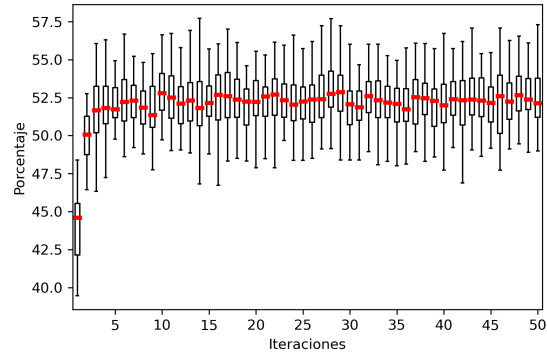
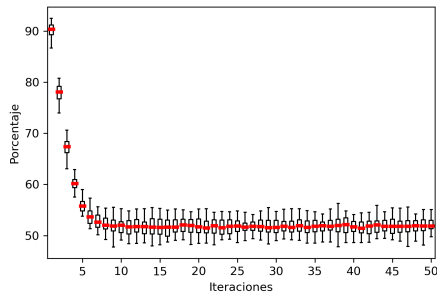
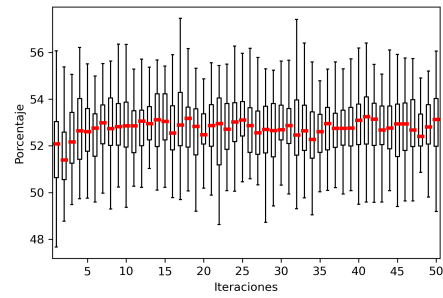


Figura 9: 32000 partículas en c3.



(a) 64000 partículas en c1.



(b) 64000 partículas en c2.

Figura 10: Gráficas de porcentaje en diferentes puntos críticos.

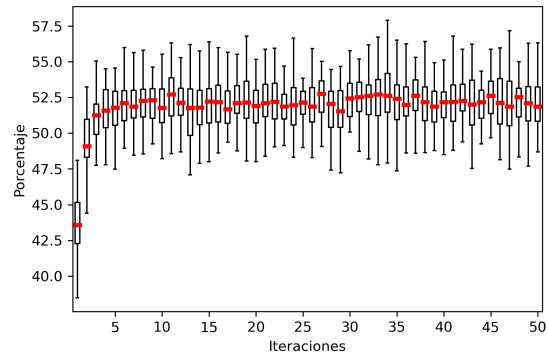
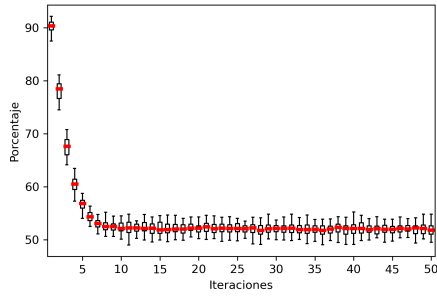
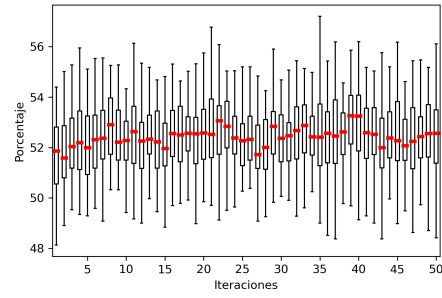


Figura 11: 64000 partículas en c3.



(a) 128000 partículas en c1.



(b) 128000 partículas en c2.

Figura 12: Gráficas de porcentaje en diferentes puntos críticos.

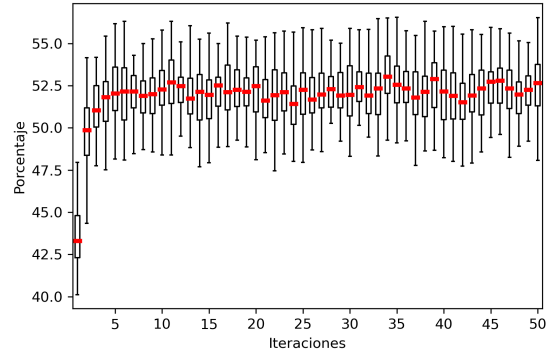
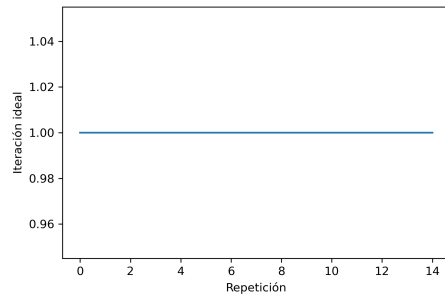


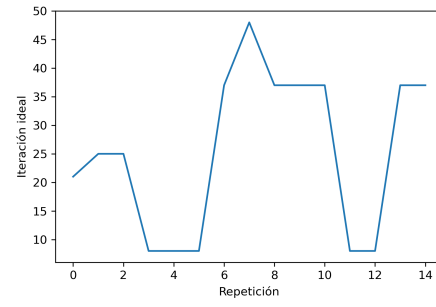
Figura 13: 128000 partículas en c3.

Cuadro 2: Prueba estadística Anova.

Partículas	F	P
16000	255.8	3.3
32000	133.7	8.1
64000	116.7	1.16
128000	170.5	3.8



(a) 16000 partículas en c1.



(b) 16000 partículas en c2.

Figura 14: Gráfica de iteración ideal por repetición en diferentes puntos críticos.

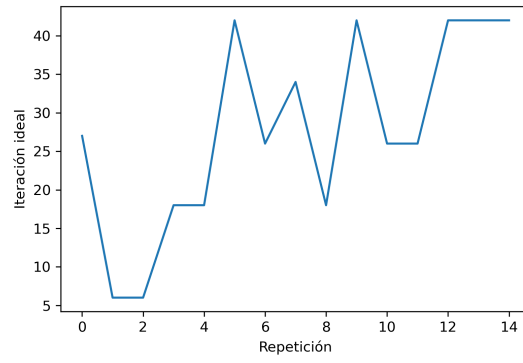


Figura 15: Gráfica de iteración ideal por repetición en 16000 partículas en c3.

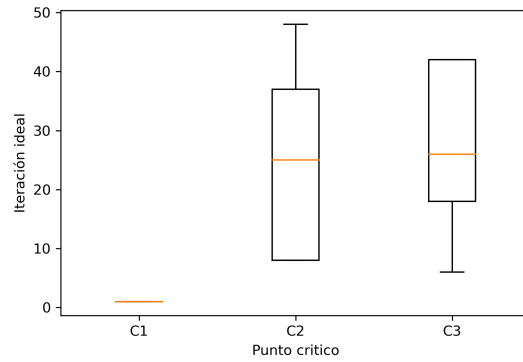
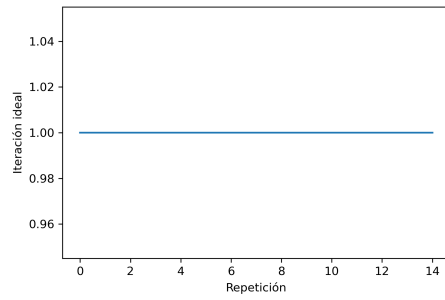
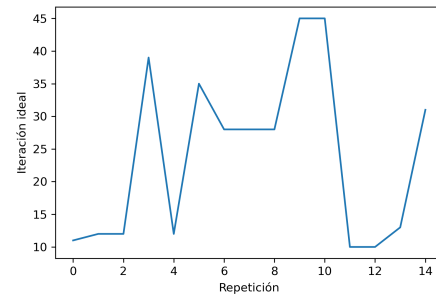


Figura 16: Gráfica caja bigote de iteración ideal con diferentes puntos críticos.



(a) 32000 partículas en c1.



(b) 32000 partículas en c2.

Figura 17: Gráfica de iteración ideal por repetición en diferentes puntos críticos.

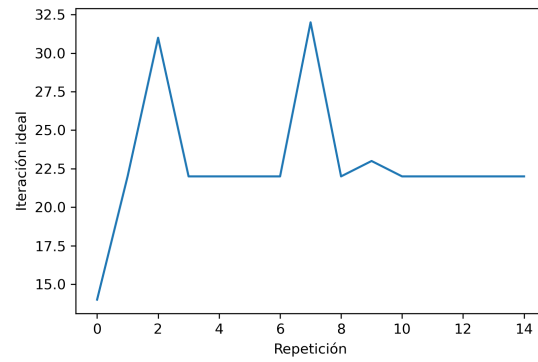


Figura 18: Gráfica de iteración ideal por repetición en 32000 partículas en c3.

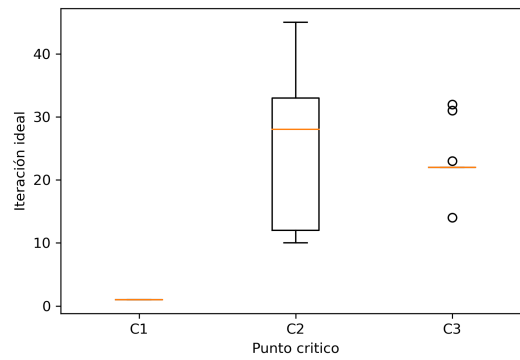
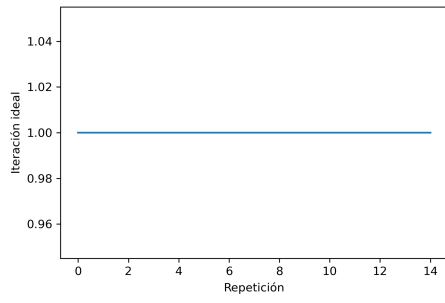
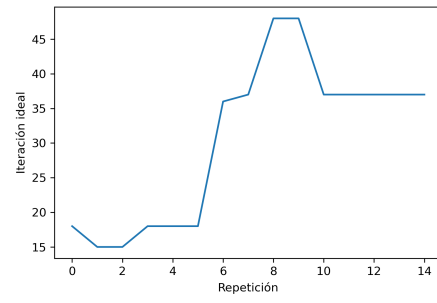


Figura 19: Gráfica caja bigote de iteración ideal en diferentes puntos críticos.





(a) 64000 partículas en c1.



(b) 64000 partículas en c2.

Figura 20: Gráfica de iteración ideal por repetición en diferentes puntos críticos.

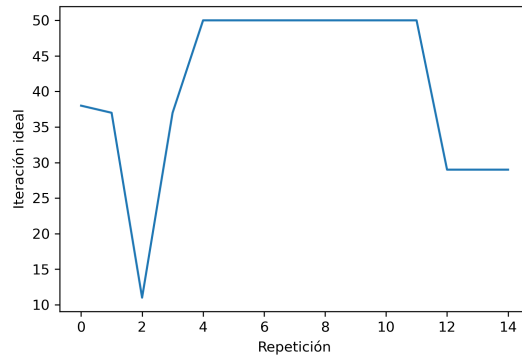


Figura 21: Gráfica de iteración ideal por repetición 64000 partículas en c3.

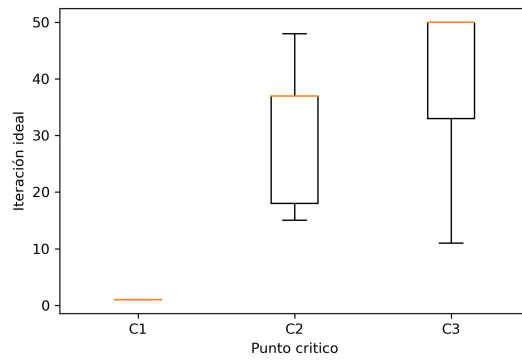
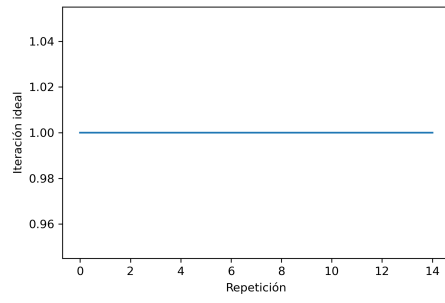
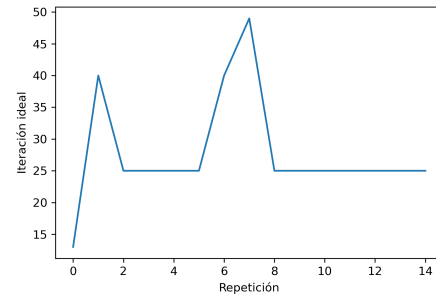


Figura 22: Gráfica caja bigote de iteración ideal con diferentes puntos críticos.



(a) 128000 partículas en c1.



(b) 128000 partículas en c2.

Figura 23: Gráfica de iteración ideal por repetición en diferentes puntos críticos.

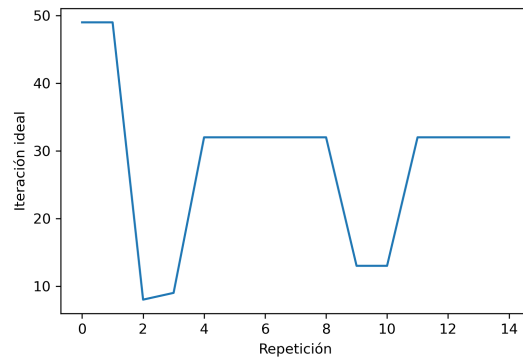


Figura 24: Gráfica de iteración ideal por repetición en 128000 partículas en c3.

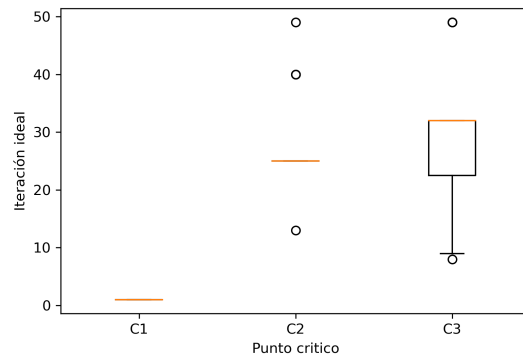


Figura 25: Gráfica caja bigote de iteración ideal con diferentes puntos críticos.

Cuadro 3: Prueba estadística Anova.

Partículas	F	P
16000	26.7	3.2
32000	39.3	2.3
64000	66.41	9.8
128000	47.34	1.7

## Referencias

- [1] *Denisse Leyva Repositorio*. URL: <https://github.com/Denisse251/Simulation/tree/main/Tarea.8>.
- [2] *Elisa Schaeffer Práctica 7*. URL: <https://elisa.dyndns-web.com/teaching/comp/par/p8.html>.
- [3] *Elisa Schaeffer Repositorio*. URL: <https://github.com/satuelisa/Simulation>.