

# Modelo FHP para dos contenedores unidos por un hueco

...

Facundo Astiz (58333), Miguel Di Luca (58460), GRUPO 3

# [1] Introducción

...

# [1] Sistema Físico

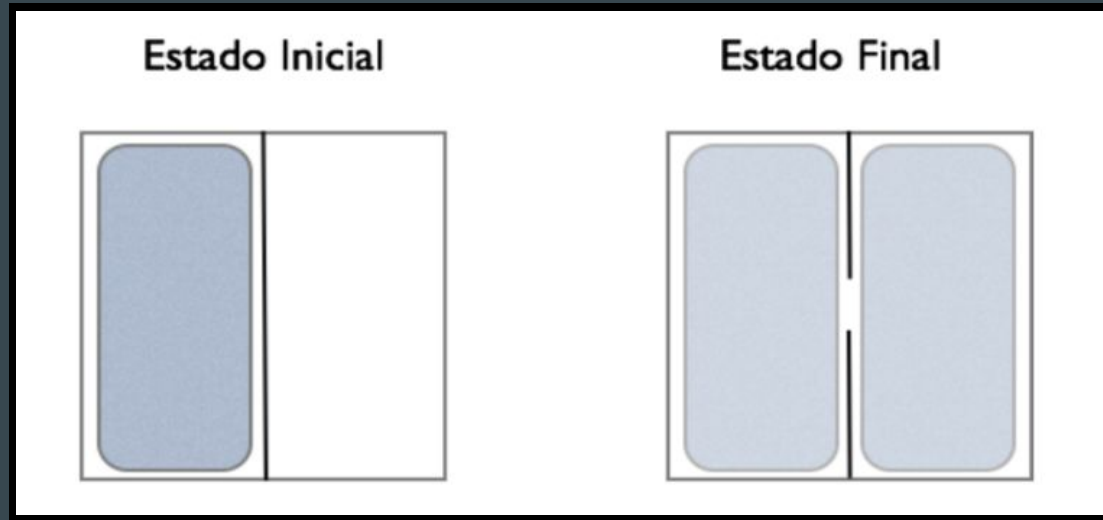
Ecuación de Navier Stokes:

$$\frac{\partial u}{\partial t} + (u \nabla) u = -\nabla P + \nu \nabla^2 u$$

- Describe el movimiento de un fluido.
- Salvo casos concretos, no es posible hallar una solución analítica.

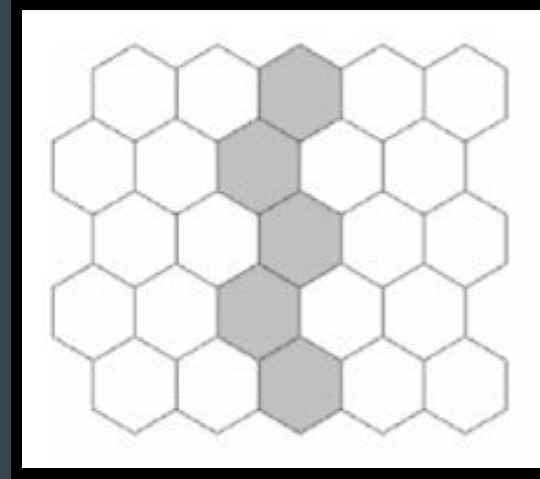
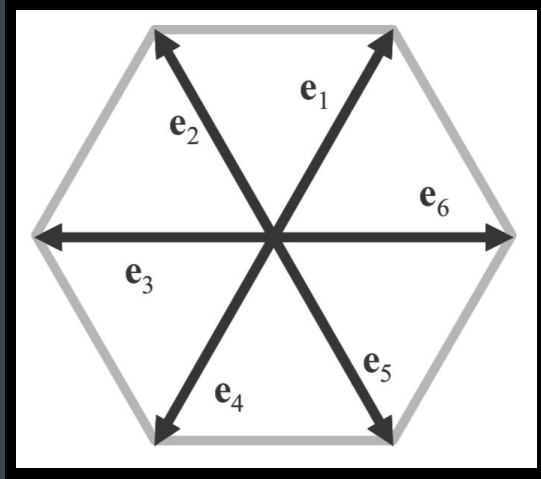
# [1] Modelo FHP

Se define el modelo “lattice-gas”, el mismo es equivalente a resolver las ecuaciones de Navier-Stokes.



# [1] Modelo FHP - Grilla

Para describir el mapa de partículas se usa una grilla hexagonal en donde pueden coexistir hasta seis partículas con velocidades distintas y con direcciones a los hexágonos adyacentes. Las velocidades y forma de la grilla se pueden ver en las siguientes figuras.



# [1] Modelo FHP - Iteración

El modelo funciona a partir de pasos o iteraciones. En cada iteración se realizan las acciones de **PROPAGACIÓN** y luego de **COLISIÓN**.

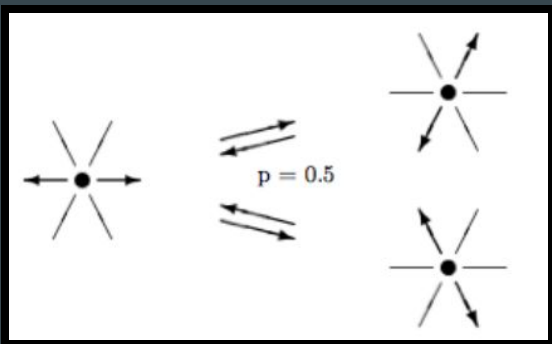
En la **PROPAGACIÓN** aplicamos las velocidades de cada partícula de la simulación. Como todas las velocidades tienen magnitud unitaria será desplazarlas a celdas adyacentes.

En la etapa de **COLISIÓN** se simula el choque entre todas las partículas que se encuentran en la misma celda. Por lo tanto se calcularán colisiones de hasta seis partículas.

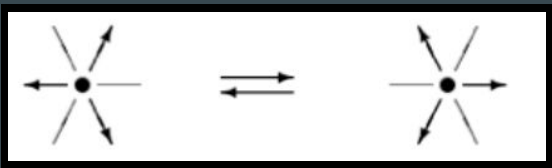
# [1] Modelo FHP - Colisiones entre partículas

La gran mayoría de las colisiones se resuelven de forma determinística pero para ser fiel al sistema físico algunas de ellas requieren que se resuelva con un factor de azar. Para eso se calcula en el momento inicial de la simulación un bit random para cada celda. De esta forma las colisiones que se resuelven estocásticamente se decidirá a partir del bit guardado en su celda.

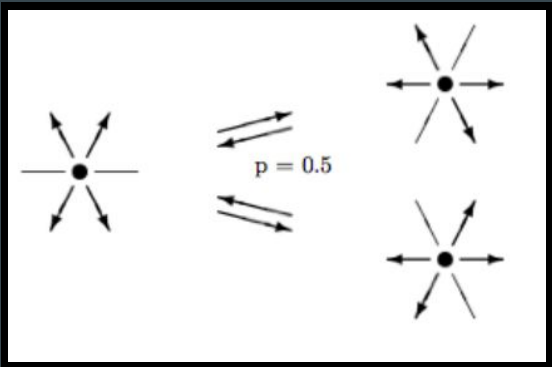
Además, de las colisiones determinísticas se descarta una gran parte porque las velocidades resultantes son equivalentes a las iniciales, como, por ejemplo, todas las colisiones entre cinco o seis partículas. Por lo tanto nos queda únicamente por resolver las colisiones que se muestran en las figuras de la siguiente diapositiva.



2 PARTÍCULAS



3 PARTÍCULAS



4 PARTÍCULAS



# [1] Modelo FHP - Condiciones de borde

Para simular los bordes o paredes marcamos ciertas celdas del mapa como sólidas. De esta forma cuando se está en la etapa de **COLISIÓN** y nos encontramos con partículas dentro de una celda con característica de sólida, ignoramos las colisiones entre las partículas y en su lugar invertimos las velocidades de cada una de ellas. De esta forma logramos de una forma simple hacer que cada una de las partículas vuelva por donde vino.

# [2] Implementación

...

## [2] Implementación - Representación de la celda

Para cada celda utilizamos un byte para su representación. Como las partículas que conviven en la misma celda tienen que tener velocidades distintas podemos representar cada dirección de velocidad con un único bit y así ocuparemos 6 bits para representar todas sus partículas. Los dos bits restantes se utilizarán para indicar si es un sólido y para representar el random que se explicó en la diapositiva de **Colisiones entre partículas**.

		Bit Value							
		128	64	32	16	8	4	2	1
VELOCIDADES	A	0	0	0	0	0	0	0	1
	B	0	0	0	0	0	0	1	0
	C	0	0	0	0	0	1	0	0
	D	0	0	0	0	1	0	0	0
	E	0	0	0	1	0	0	0	0
	F	0	0	1	0	0	0	0	0
	SÓLIDO	S	0	1	0	0	0	0	0
	RANDOM	R	1	0	0	0	0	0	0

## [2] Implementación - Propagación

Debido a como se representa cada celda resulta muy sencillo superponer velocidades; lo único que hay que hacer es sumarlas. Por ejemplo si tenemos las velocidades  $A = [0\ 0\ 0\ 0\ 0\ 0\ 1]$  y  $B = [0\ 0\ 0\ 0\ 0\ 1\ 0]$  y queremos representar las velocidades A y B en una celda, lo único que tenemos que hacer es  $A + B = [0\ 0\ 0\ 0\ 0\ 1\ 1]$ .

Por lo tanto, para propagar velocidades creamos una copia del mapa sin preservar las velocidades. Es decir, solo copiando el bit de sólido y el de random, y poniendo 0s en todos los bits de velocidad para cada una de las celdas.

Luego, para cada celda del mapa viejo encontramos las velocidades y se las sumamos a la celda de la copia del mapa que se encuentre en esa dirección.

Cuando terminemos tomamos la copia como nuestro nuevo mapa.

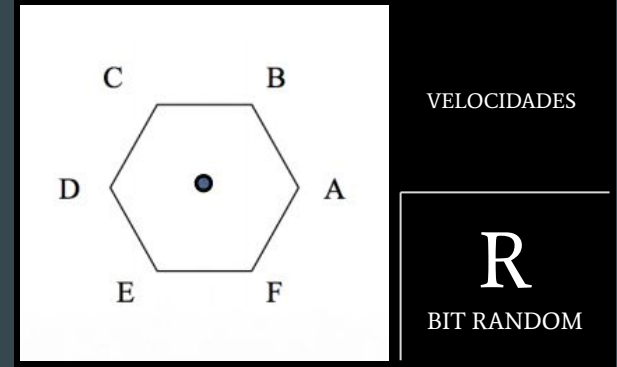
## [2] Implementación - Colisiones entre partículas

Resolver las colisiones también resulta sencillo gracias a la forma en la que guardamos las partículas.

La gran mayoría de las colisiones debemos ignorarlas porque producen un mismo resultado de velocidades. Solo nos resta calcular las colisiones restantes por medio de una tabla o *switch*. Las tablas se muestran en las siguientes diapositivas según la cantidad de partículas involucradas.

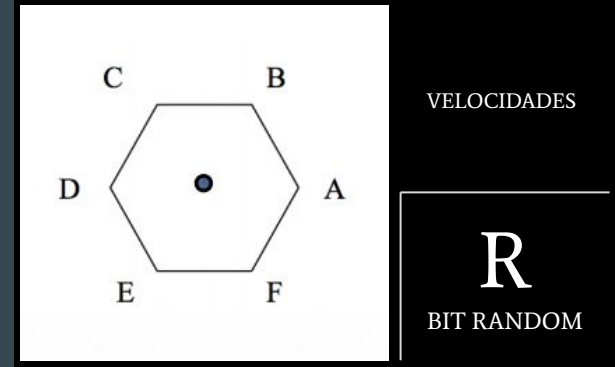
# 2 PARTÍCULAS

Entrada	Salida
$A + D$	$B + E$
$B + E$	$C + F$
$C + F$	$A + D$
$A + D + R$	$C + F$
$B + E + R$	$A + D$
$C + F + R$	$B + E$



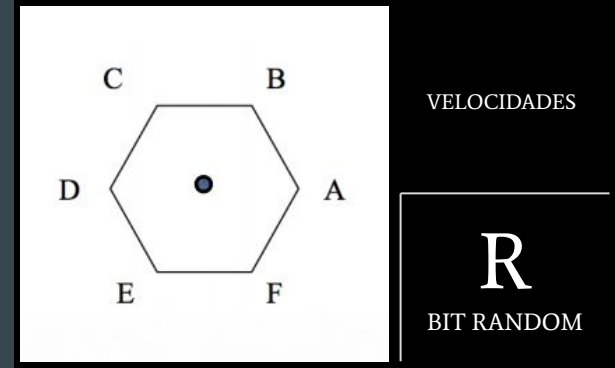
# 3 PARTÍCULAS

Entrada	Salida
$A + C + E$	$B + D + F$
$B + D + F$	$A + C + E$
$A + C + E + R$	$B + D + F + R$
$B + D + F + R$	$A + C + E + R$



# 4 PARTÍCULAS

Entrada	Salida
$C + B + E + F$	$C + D + A + F$
$C + B + E + F + R$	$B + A + D + E$
$C + D + A + F$	$E + B + D + A$
$C + D + A + F + R$	$E + B + C + F$
$D + E + B + A$	$C + F + D + A$
$D + E + B + A + R$	$C + F + B + E$





## [2] Implementación - Condiciones de borde

Para resolver las colisiones contra sólidos usamos también un sistema de tablas. Para todas las celdas que tengan el bit de sólido prendido, vamos a reemplazarla por un byte espejado en las velocidades. Es decir, dejamos el bit sólido prendido y para cada velocidad encontramos su opuesta.

# [2] Implementación - Arquitectura

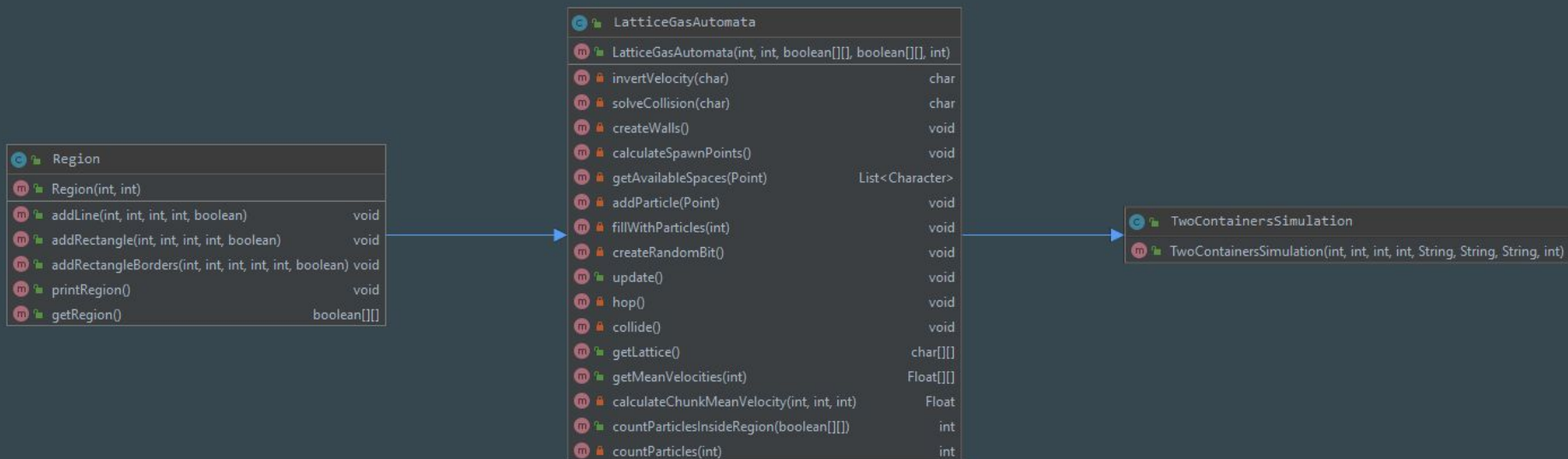
## Simulación

- Para la implementación de la simulación se utilizó Java.
- La simulación imprime archivos con los datos obtenidos.

## Procesamiento de datos

- Mediante Python se realizó una implementación propia, específica para el problema.
- Esta procesa los archivos generados por la simulación, muestra el desarrollo de la misma y genera los gráficos.

## [2] Implementación - Diagrama UML

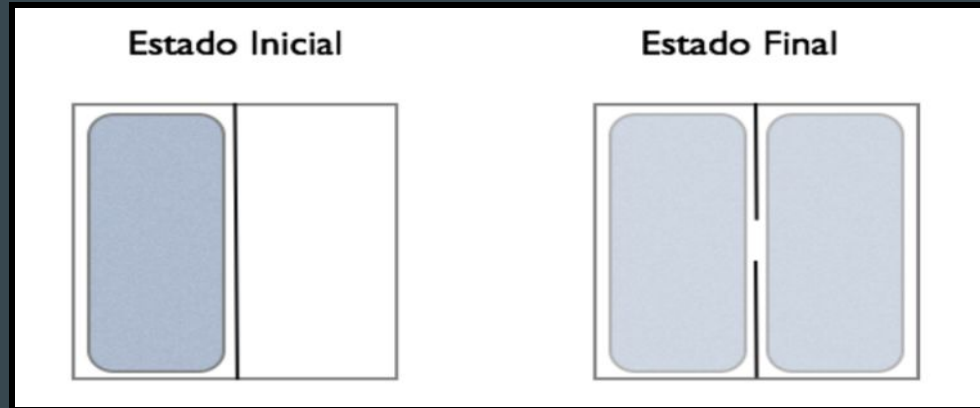


# [3] Simulaciones

...

# [3] Simulación - Sistema de estudio

- En la simulación se estudiará un sistema en el cual se tiene una grilla de 200 x 200 celdas.
- Inicialmente, la grilla se encuentra dividida en dos mediante una pared y todo el gas se encuentra en el contenedor izquierdo.
- Luego, se abre un hueco de 50 celdas en la mitad de la pared, conectando de esta forma los dos contenedores.



## [3] Simulación - Parámetros

- El tamaño y la forma del mapa, se mantienen constantes. Aunque se podría experimentar con otras formas y otros tamaños, ya que la implementación lo contempla.
- Se realizaron múltiples simulaciones, variando únicamente la cantidad de partículas contenidas inicialmente en el contenedor izquierdo, las cantidades a analizar serán 2000, 3000, 5000, 20000, 50000 y 100000 partículas.

## [3] Simulación - Ejecución

- Se ejecutó, para cada  $N$ , veinte veces la simulación, guardando para cada tiempo la cantidad de partículas en cada contenedor
- A partir de estos datos se calculó la media y el desvío estándar.

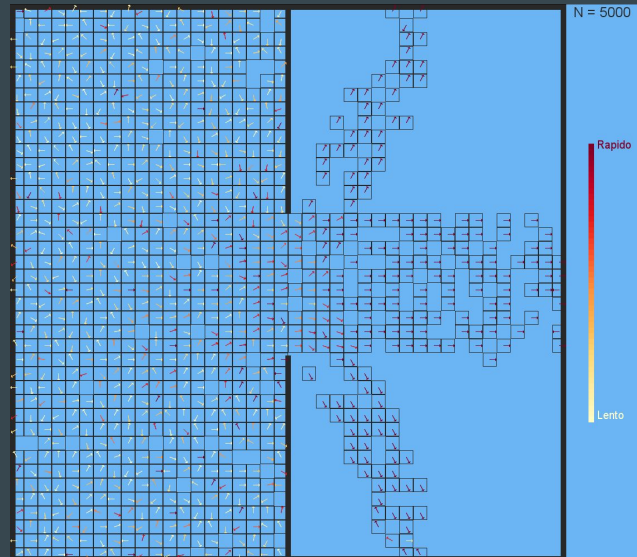
# [4] Resultados

...



## [4] Resultados - Simulación

Para mostrar la simulación se dividió el mapa en bloques de 5 x 5 celdas y se muestra mediante una flecha la dirección y la velocidad promedio (espacial) de ese bloque en cada instante. La velocidad se encuentra representada por el color de la flecha. En la siguiente imagen se pueden apreciar los bloques.



## [4] Resultados - Simulación

A continuación se muestra la cantidad de partículas en ambos contenedores al final de la simulación para veinte ejecuciones distintas con 5000 partículas.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cant idad Izqui erda	2521	2472	2522	2565	2497	2629	2542	2550	2541	2563	2558	2595	2561	2554	2563	2533	2597	2538	2538	2512
Cant idad Dere cha	2479	2528	2478	2435	2503	2371	2458	2450	2459	2437	2442	2405	2439	2446	2437	2467	2403	2462	2462	2488

# [4] Resultados - Animación

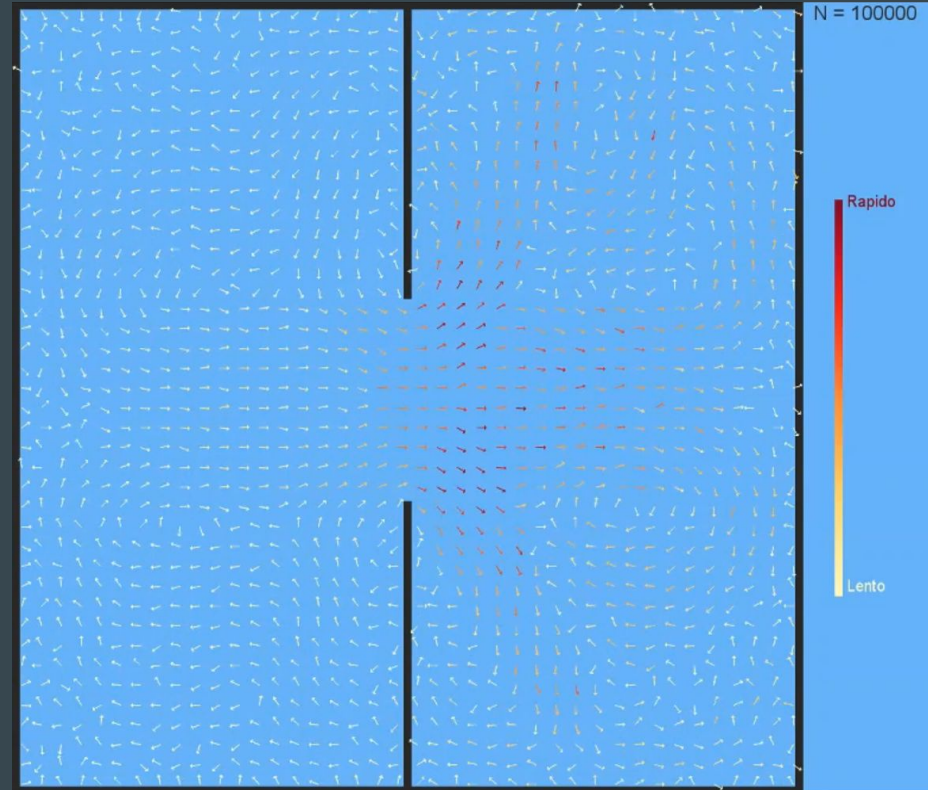
Link a animación completa:

100000 partículas:

<https://www.youtube.com/watch?v=TCwIQ9LxB0s>

5000 partículas:

<https://www.youtube.com/watch?v=KlcD4SNd7eY>

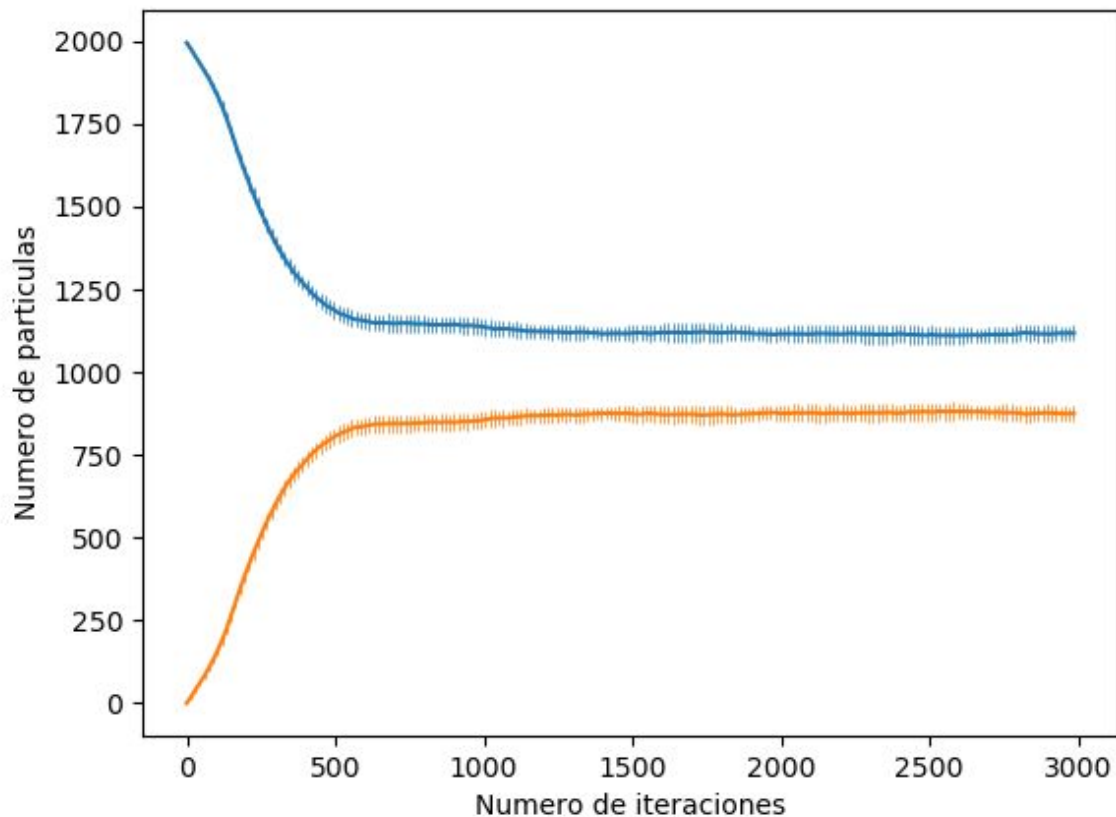


## [4] Resultados - Curvas de cantidad de partículas

En las siguientes diapositivas mostraremos el gráfico de la cantidad de partículas en cada contenedor en función de la cantidad de iteraciones transcurridas.

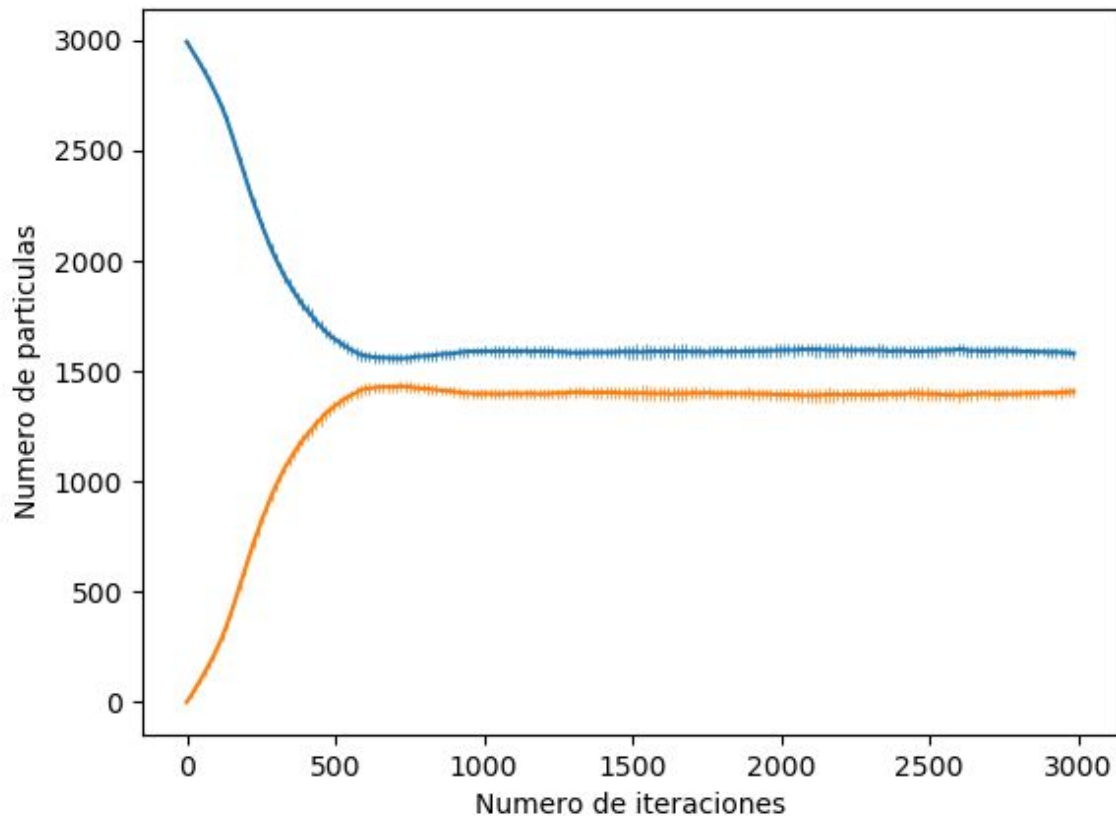
Las simulaciones se ejecutan 20 veces para poder sacar una media y desvío estándar.

# 2000 PARTÍCULAS



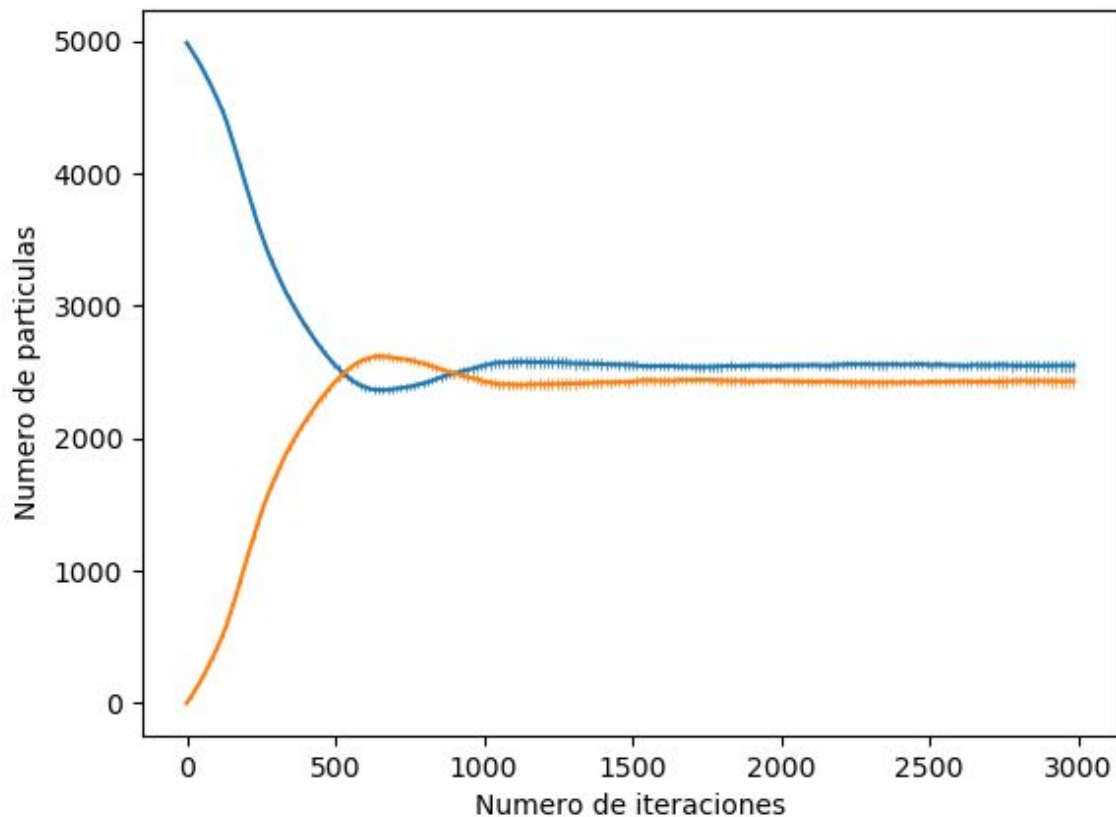
Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

# 3000 PARTÍCULAS



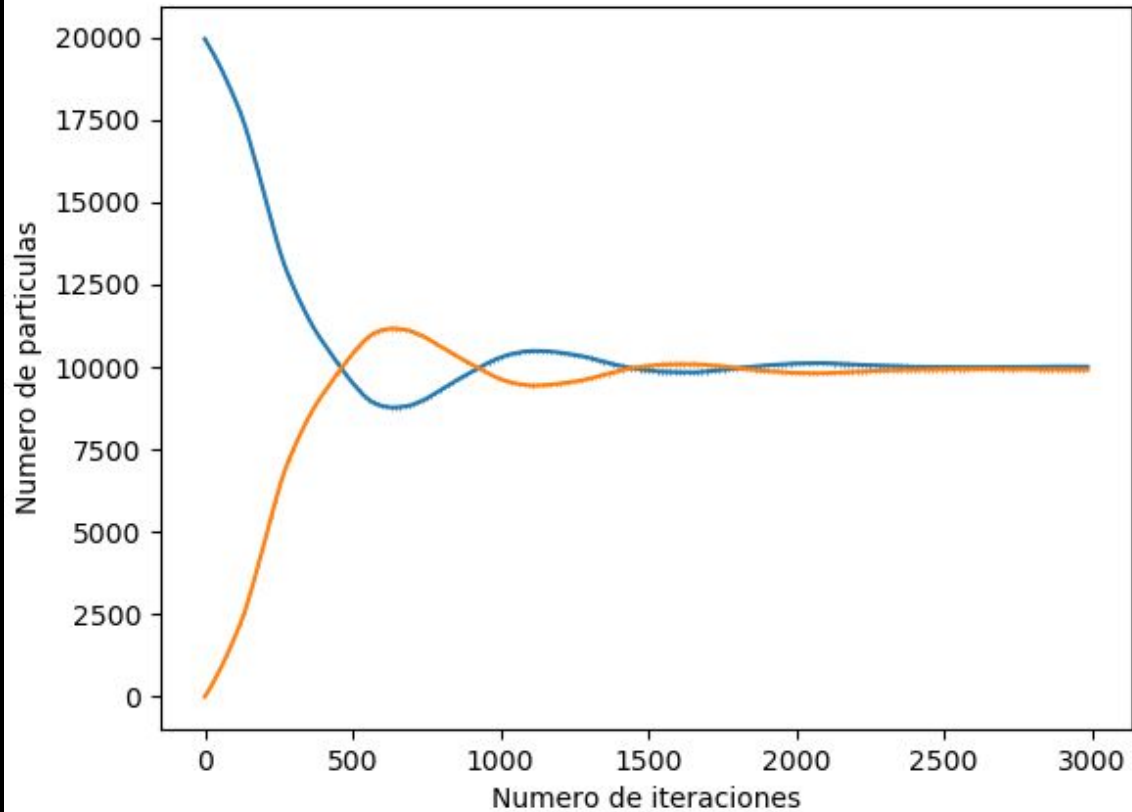
Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

# 5000 PARTÍCULAS



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

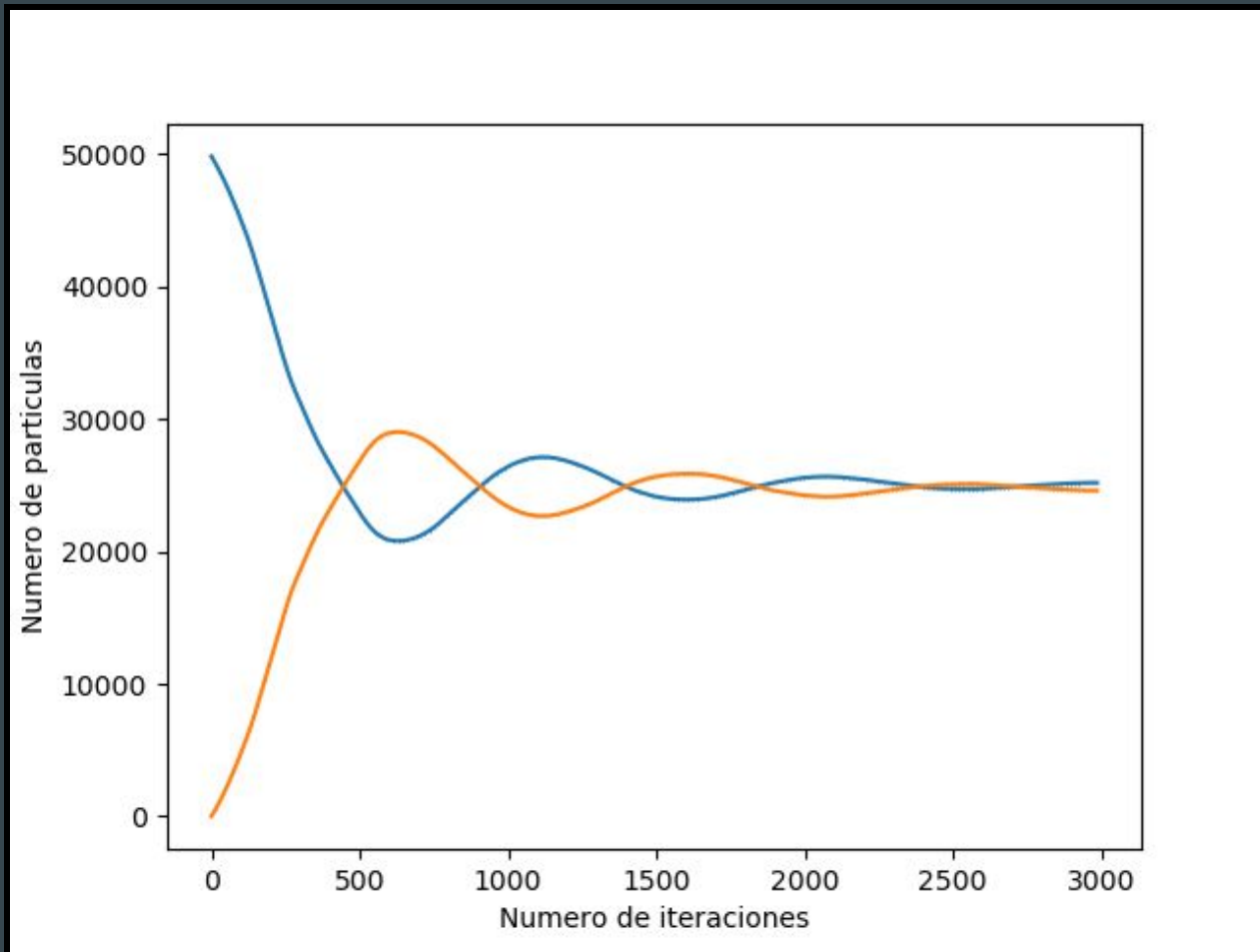
# 20000 PARTÍCULAS



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

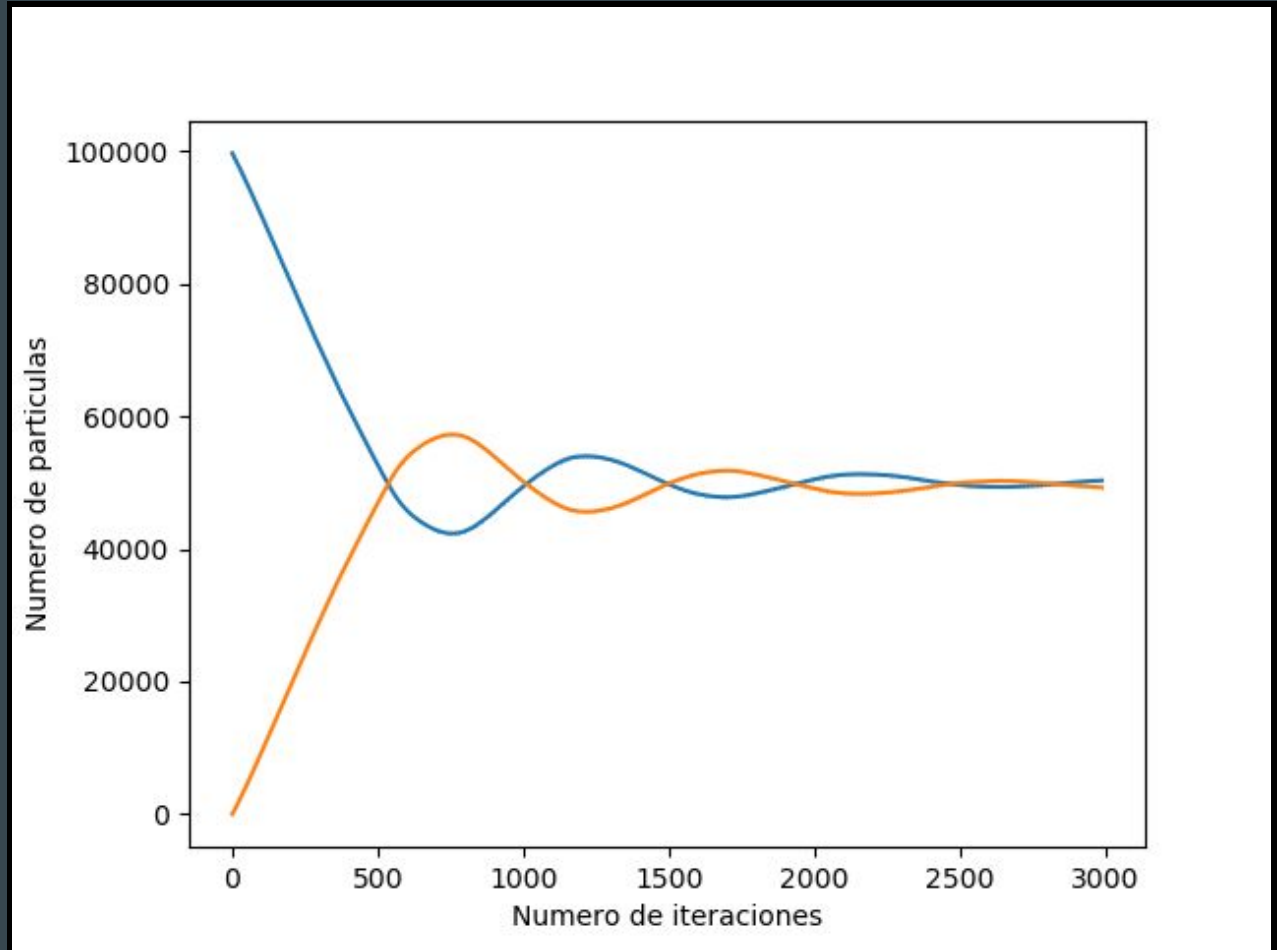


# 50000 PARTÍCULAS



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

# 100000 PARTÍCULAS



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

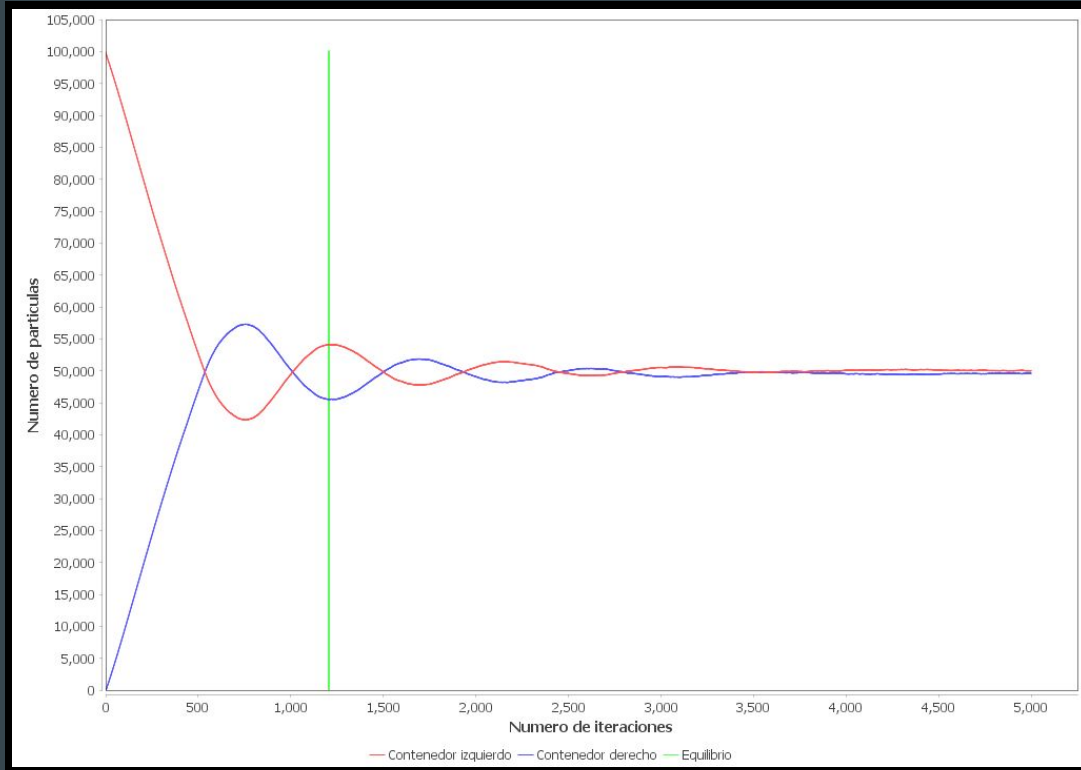
## [4] Resultados - Criterio de equilibrio

El comportamiento de la densidad de partículas sufre rebotes, especialmente cuanto más grande es la cantidad de partículas involucradas. Además, hay que tener en cuenta que la máxima cantidad de partículas desplazadas de un contenedor a otro en cada rebote es cada vez menor.

Como criterio de equilibrio, encontramos el momento en el que la diferencia máxima de partículas entre los contenedores entre dos raíces sea menor a un valor  $f$  previamente establecido. En el caso en el que no se alcance nunca una raíz, tomaremos como equilibrio el primer momento en el que  $f$  supere la diferencia porcentual de partículas.

El criterio se muestra con figuras en las siguientes diapositivas.

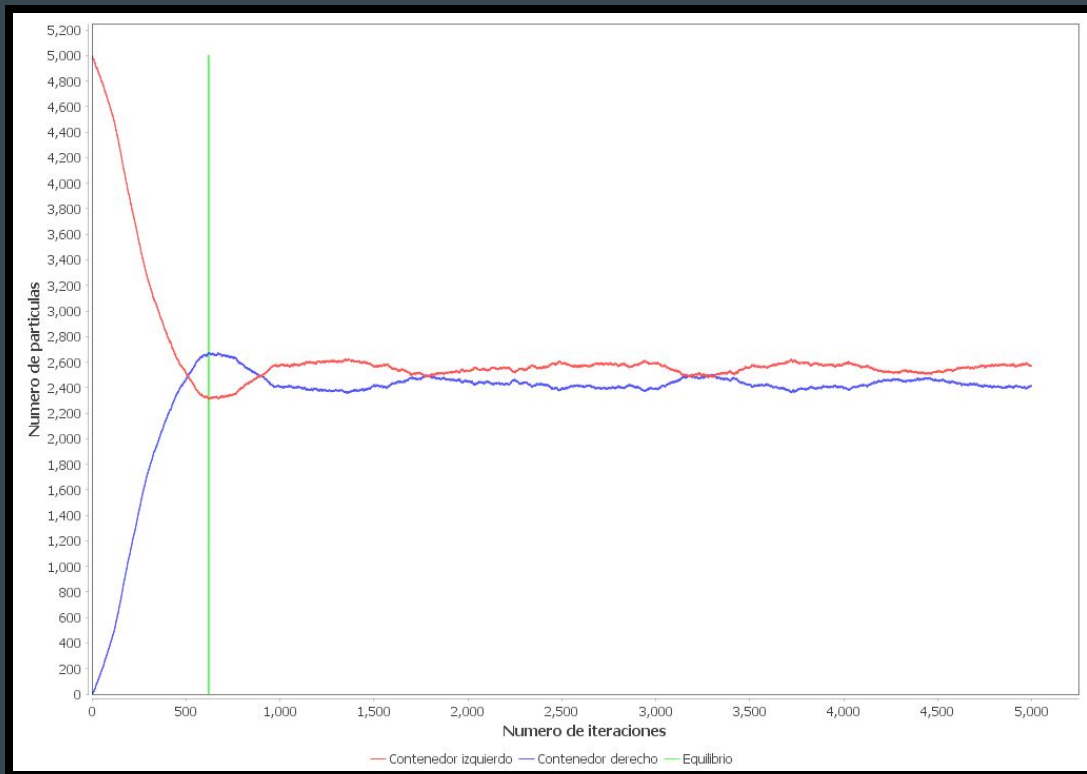
## [4] Resultados - Criterio de equilibrio



$$f = 5\%$$

Partículas  
= 100.000

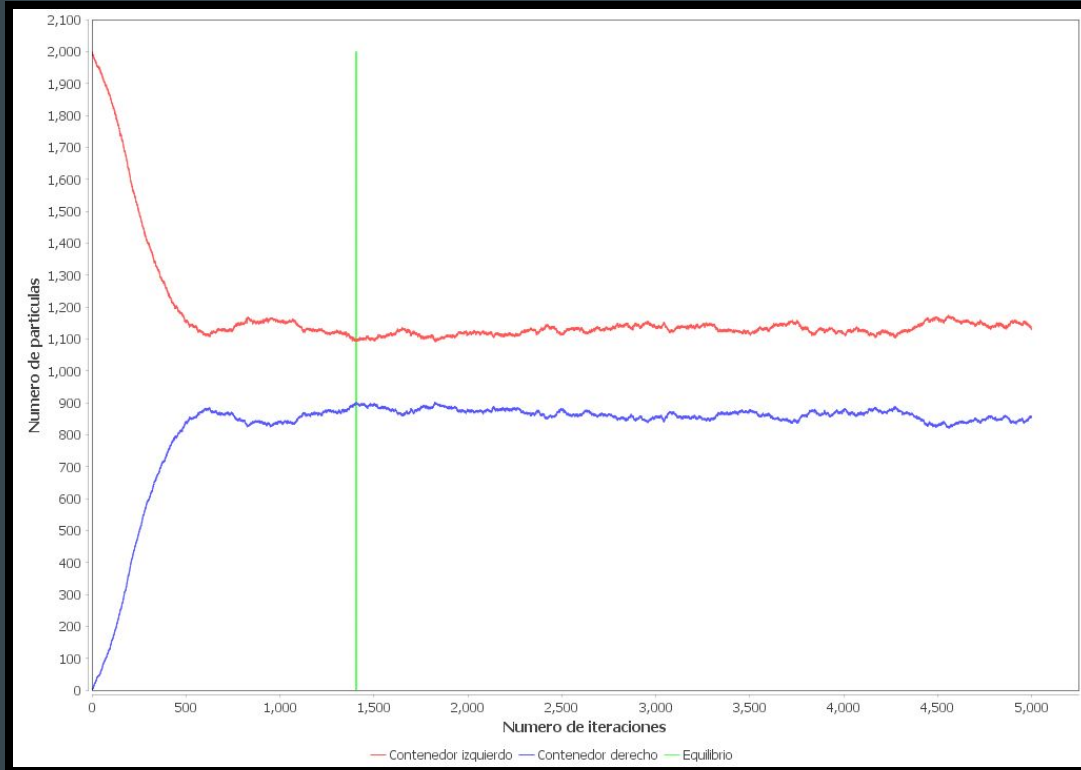
## [4] Resultados - Criterio de equilibrio



$$f = 5\%$$

Partículas  
= 5.000

## [4] Resultados - Criterio de equilibrio



$$f = 5\%$$

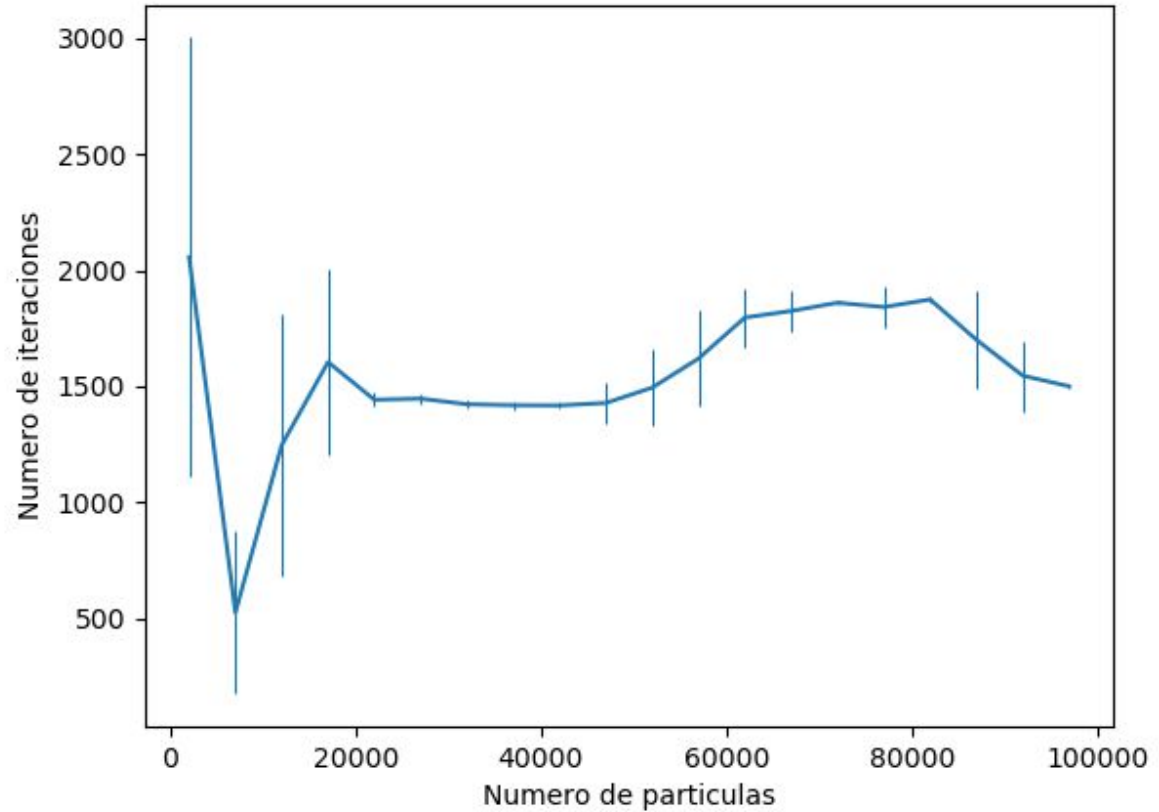
Partículas  
= 2.000

## [4] Resultados - Tiempo para alcanzar el equilibrio

A continuación se grafica el tiempo requerido para alcanzar el equilibrio. En el eje vertical se representa la cantidad de iteraciones y en el horizontal la cantidad de partículas del sistema.

El desvío estándar tan grande para algunos valores es consecuencia del criterio que elegimos. Hay momentos en el que los máximos tienen valores muy parecidos a  $f$  y por lo tanto en algunas ejecuciones el  $f$  supera a la diferencia y en otras recién se supera en el máximo consecutivo.

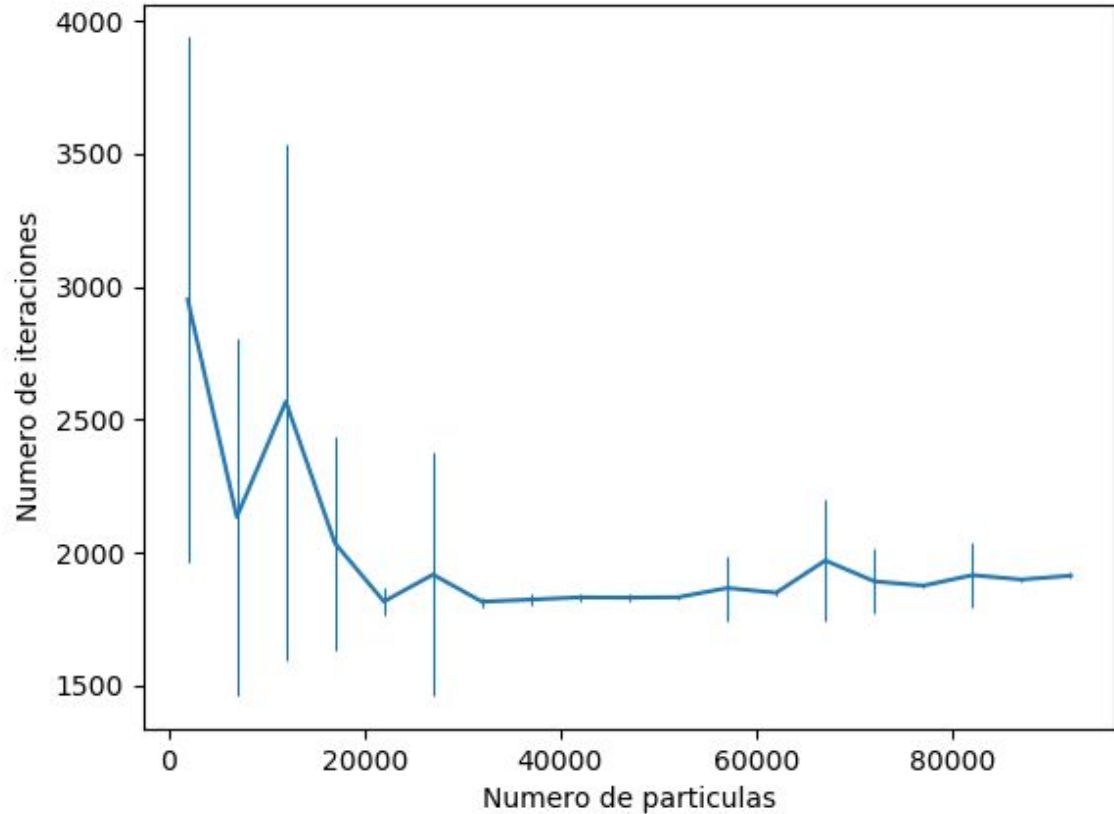
$$f = 5\%$$



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.



$$f = 2.5\%$$



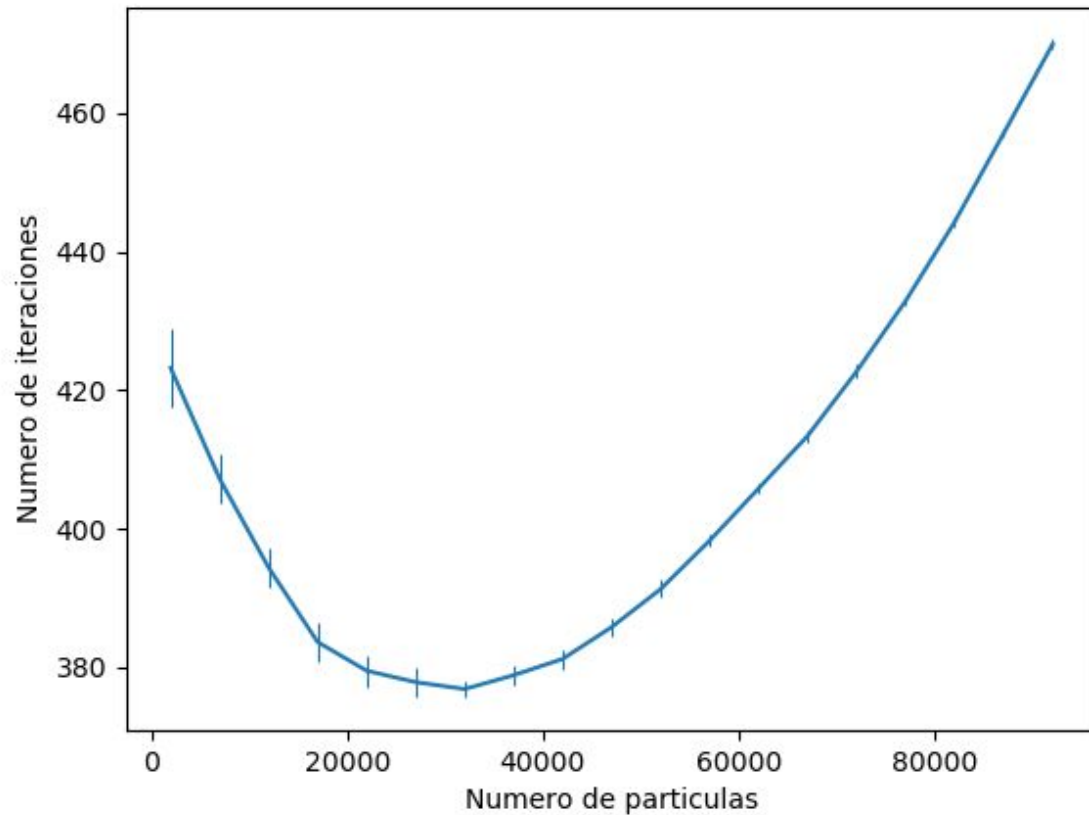
Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

## [4] Resultados - Otro criterio de equilibrio

Probamos, también, con un criterio de equilibrio más simple. Encontrar el momento en el que la diferencia porcentual de partículas entre cada contenedor sea menor a un  $f$  establecido.

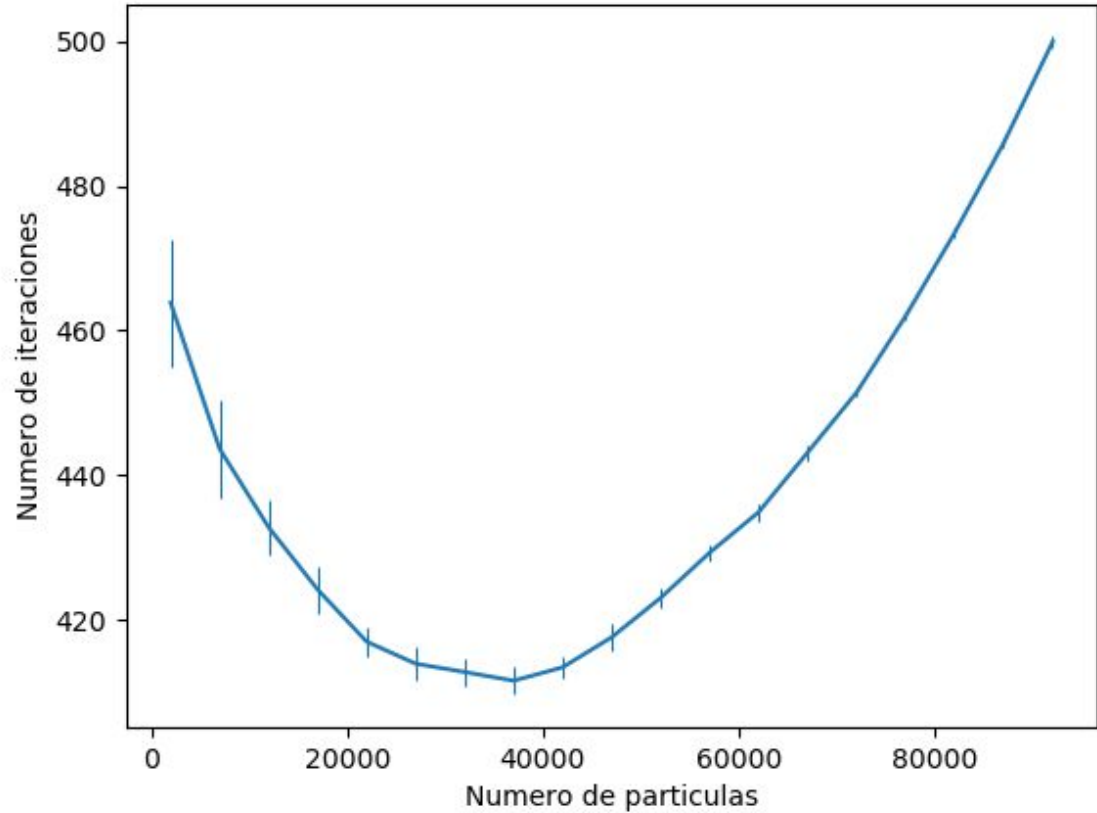
A continuación mostramos los gráficos obtenidos.

$$f = 5\%$$



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

$$f = 2.5\%$$



Se ejecutó la simulación 20 veces para poder estimar mejor la media y desvío estándar.

# [5] Conclusiones

Encontramos que una de las ventajas del método es que el tiempo de ejecución no aumenta si agregamos partículas debido a que las operaciones se realizan a las celdas. Por lo tanto podemos llenar el mapa de partículas hasta saturar y conseguir un resultado en el mismo tiempo.

El contenedor izquierdo podemos cargarlo con más de 115000 partículas ocupando todas las velocidades. Esto se puede simular, por lo explicado anteriormente, sin pérdida en cuanto al tiempo de ejecución. Por lo tanto consideramos que el comportamiento extraño cuando utilizamos 2000 y 3000 partículas se explica por generar una cantidad mucho menor de partículas a la máxima posible.

En cuanto al tiempo requerido para encontrar el equilibrio vimos que para el primer criterio se mantiene constante a excepción de una subida en cantidad de iteraciones alrededor de las 70000 partículas. Pero para el segundo criterio encontramos que alcanzamos el equilibrio más rápido cuando estamos alrededor de las 35000 partículas.