

# Dinámica Peatonal

## Trabajo Práctico Nro. 6

Badi Leonel, Buchhalter Nicolás Demián y Meola Franco  
Román

9 de junio de 2016

Grupo 3

# Fundamentos

## Introducción

- Vamos a simular el egreso de personas de una habitación
- Para ello utilizaremos el *Contractile Particle Model*
- No se utilizaron plataformas de simulación peatonal

# Implementación

## Generación de las partículas

- Posiciones  $(x, y)$  aleatorias
  - Verificando que no se superpongan
  - Intentando hasta 10000 veces por partícula para obtener una ubicación válida
- $N = 300$
- $r_{min} = 0,15m$
- $r_{max} = 0,32m$
- $v_d^{max} = 1,55 \frac{m}{s}$

# Simulación

Variables relevantes del sistema

- **Parámetros de la habitación:**

- Habitación cuadrada de  $20m \times 20m$
- Puerta central de  $1,2m$  de apertura

- **Parámetros del Contractile Particle Model:**

- $\tau = 0,5s$
- $\beta = 0,9$

# Simulación

## Variables relevantes de la simulación

- $t$  [s]: tiempo en segundos a visualizar
- $dt$  [s]: tiempo en segundos del paso de la simulación
- $k$ : relación entre cantidad de pasos simulados y escritos

# Simulación

## Algoritmo de simulación

```
public void simulate(double t, double dt, int k){  
    writeFrame(0);  
    int framesWritten = 1;  
    double totalTimeSimulated = 0;  
    moveSystem(dt);  
    totalTimeSimulated += dt;  
    while(totalTimeSimulated < t){  
        for(int i = 0; i < k; i++){  
            moveSystem(dt);  
            totalTimeSimulated += dt;  
        }  
        writeFrame(framesWritten++);  
    }  
}
```

Código 1: Algoritmo de simulación

# Simulación

## Detalles de implementación

- Para los valores de  $N$  que vamos a analizar se obtuvieron tiempos de procesamiento muy bajos
- Por lo tanto no se necesitó contar con el *Cell Index Method*
- Se implementó un modelo que contempla una lista de *targets* para cada una de las partículas

# Implementación

## Visualización

- La simulación y la visualización son independientes
- El algoritmo de simulación escribe un archivo .tsv con los siguientes datos:
  - $(x, y)$
  - $r$
  - Color RGB para indicar las velocidades, donde R es la componente en el eje Y y G es la componente en eje X
- Por último, se carga en Ovito el archivo de salida .tsv para realizar la visualización

# Resultados

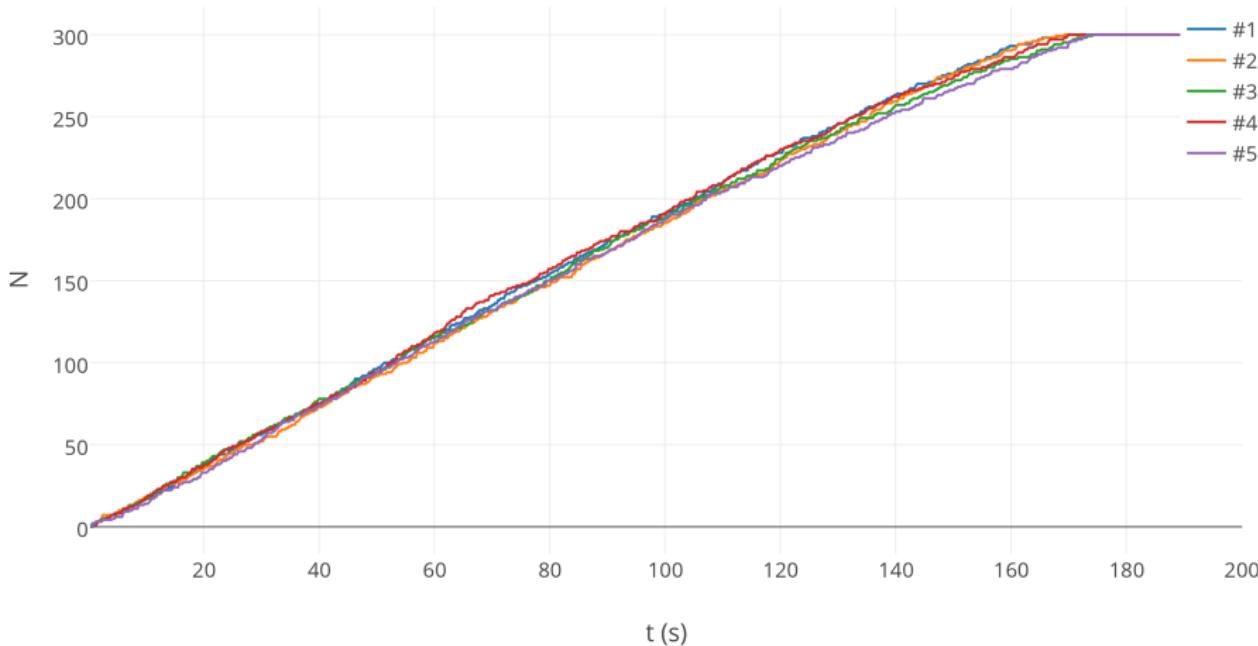
## Variables relevantes

- **Parámetros de la simulación:**

- $k = 5$
- $dt = 0,05s$

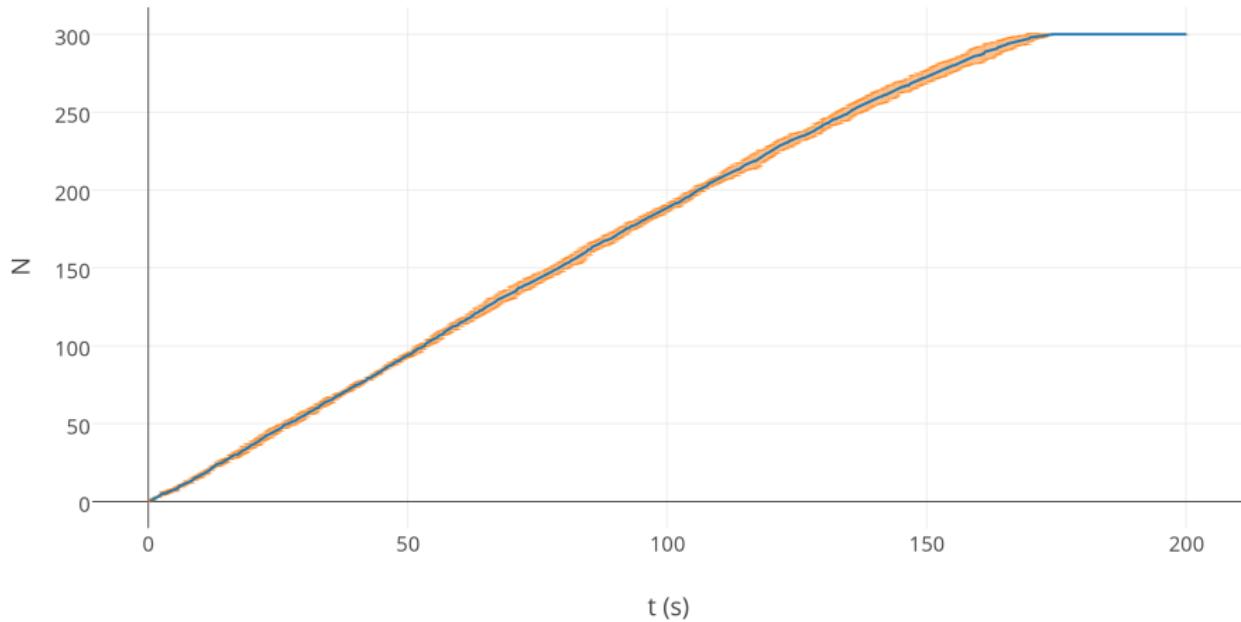
# Gráfico de número de evacuados en función del tiempo

Para  $N = 300$ ,  $r_{min} = 0,15m$ ,  $r_{max} = 0,32m$  y  $v_d^{max} = 1,55 \frac{m}{s}$



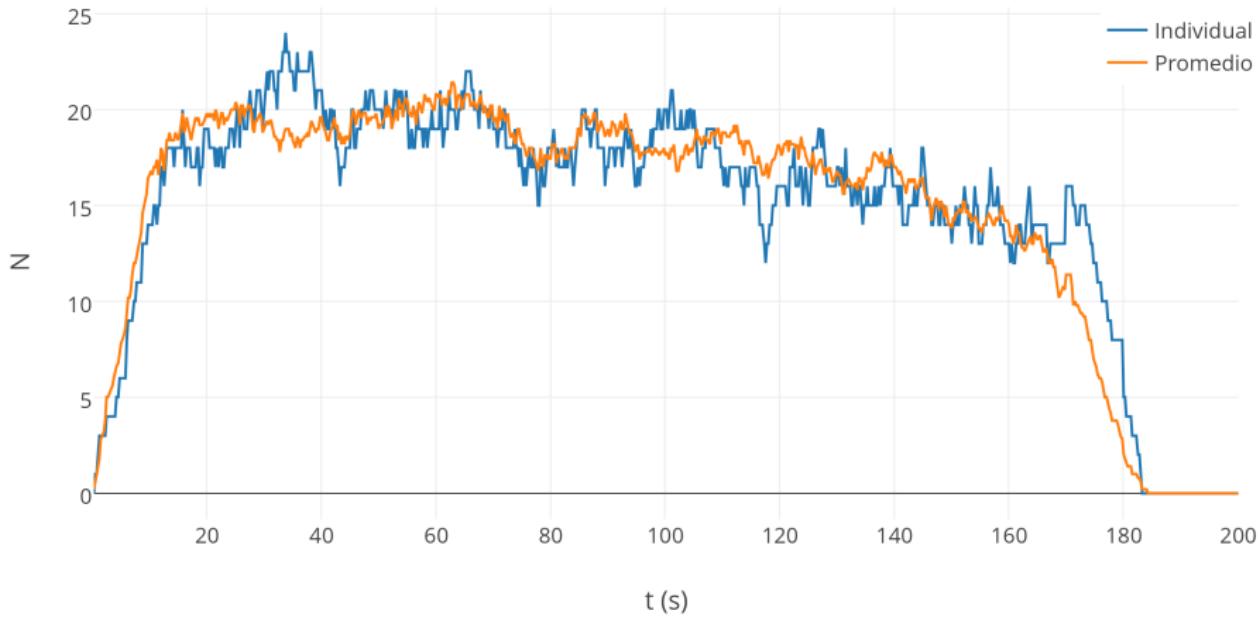
# Gráfico del comportamiento promedio del sistema

Para  $N = 300$ ,  $r_{min} = 0,15m$ ,  $r_{max} = 0,32m$  y  $v_d^{max} = 1,55\frac{m}{s}$



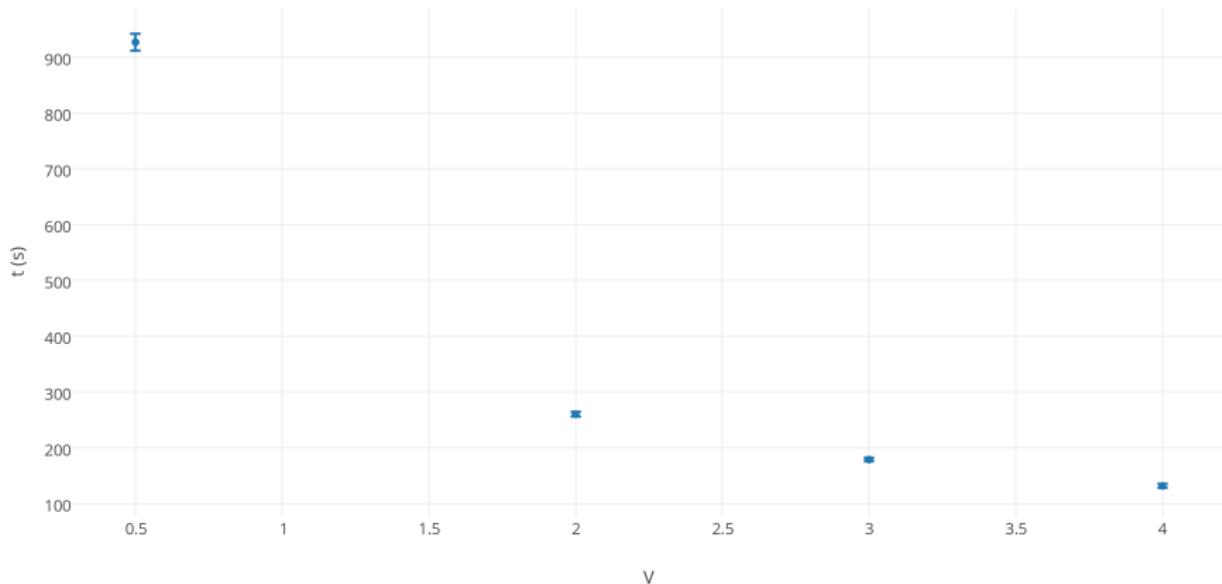
# Gráfico de la evolución del caudal en función del tiempo

Para  $N = 300$ ,  $r_{min} = 0,15m$ ,  $r_{max} = 0,32m$ ,  $v_d^{max} = 1,55\frac{m}{s}$  y Ventana = 10s



# Gráfico de $v_d^{max}$ en función del tiempo de evacuación

Para  $N = 300$ ,  $r_{min} = 0,15m$ ,  $r_{max} = 0,32m$  y 3 simulaciones en cada caso



# Animación de la simulación

Para  $N = 300$ ,  $r_{min} = 0,15m$ ,  $r_{max} = 0,32m$  y  $v_d^{max} = 1,55\frac{m}{s}$

# Conclusiones

- Se logró representar el movimiento de personas en estado no competitivo usando el *Contractile Particle Model*
- A medida que se aumenta la velocidad, se puede ver como disminuye el tiempo de salida. En este caso no se obtuvo *faster is slower*. Una de las posibles causas de esto es que el movimiento de los agentes es no competitivo.
- Implementación simple donde cada partícula o agente es independiente.

# Gracias