



Trabajo Práctico Nro. 5

Dinámica Peatonal

Grupo 8
Baliarda, Gonzalo
Pérez, Ezequiel

1C 2023

Introducción





Interacción de peatones

- Peatones con radio variable

$$r = \frac{r_{max}}{\left(\frac{\tau}{\Delta t}\right)}$$

- Si el peatón colisiona

$$r = r_{min}$$

- Posición de un peatón

$$x(t) = x(t - \Delta t) + v(t)\Delta t$$

Interacción de peatones

- Peatones con velocidad variable
- Si el peatón no está en contacto

$$v = v_d^i = v_{dmax} \left[\frac{(r^i - r_{min})}{(r_{max} - r_{min})} \right]^\beta e_{target}$$

$$e_{target} = \frac{(x_{target} - x^i)}{|x_{target} - x^i|}$$

- Si el peatón está en contacto

$$v = v_e^i = v_{dmax} \frac{(\sum_j e^{ij})}{|\sum_j e^{ij}|}$$

$$e^{ij} = \frac{(x^i - x^j)}{|x^i - x^j|}$$

Implementación



Arquitectura del código

```
class Particle {
    double minRadius
    double maxRadius
    double radius
    double v
    double vdMax
    Vector position
    double beta

    void updateRadius(double dt, boolean isInContact)
    void updateVelocity(boolean isInContact)
    boolean updatePosition(double dt, List particlesInContact, List wallsInContact)
}
```

```
class Vector {
    double x
    double y
}
```

```
class PedestrianSystem {
    List particlesOutside
    double time
    double dt
    int particlesOutsideRoom
    CellIndexMethod cim

    int nextStep()
    boolean hasNextStep()
}
```

```
enum Walls {
    LEFT,
    RIGHT,
    TOP,
    BOTTOM

    Vector getDirection()
}
```

Actualización de estado

```
int nextStep() {
    Map particlesInContact = cim.computeNeighbourhoods()
    Map wallsInContact = computeWallsInContact()

    for particle in particles {
        boolean isParticleInContact = mapsContainsKey(particle)

        p.updateRadius(dt, isParticleInContact)

        p.updateVelocity(isParticleInContact)
    }

    for particle in particles {
        boolean wentOut = p.updatePosition(dt, particlesInContact, wallsInContact)
        particlesOutsideRoom += wentOut
    }

    int previousSize = particles.size()
    particles = removeParticlesOutsideSimulation()
    time += dt

    return previousSize - particles.size()
}
```

```
boolean hasNextStep() {
    return particles.size() > 0
}
```

Actualización de radio y velocidad

```
void updateRadius(double dt, boolean isInContact) {  
    if isInContact {  
        radius = minRadius  
    } else {  
        radius += maxRadius / (tau / dt)  
    }  
}
```

```
void updateVelocity(boolean isInContact) {  
    if isInContact {  
        v = vdMax  
    } else {  
        v = vdMax * pow( (radius - minRadius)/(maxRadius - minRadius) , beta)  
    }  
}
```


Actualización de posición

```
boolean updatePosition(double dt, List particlesInContact, List wallsInContact) {
    Vector particleDirection

    if isInContact {
        for p in particlesInContact {
            particleDirection += (p.position - this.position)/abs(p.position - this.position)
        }

        for w in wallsInContact {
            particleDirection += (w.getDirection() - this.position)/abs(w.getDirection() - this.position)
        }
    } else {
        double leftExit = getBoxLength() / 2 - L / 2
        double[] decisionInterval = {leftExit + 0.2*L, leftExit + 0.8*L}
        double xTarget = x, yTarget = 0;

        if (x < decisionInterval[0] || x > decisionInterval[1]) {
            xTarget = decisionInterval[0] + random() * (decisionInterval[1] - decisionInterval[0])
        }

        particleDirection = directionToTarget(xTarget, yTarget)
    }

    double direction = atan2(particleDirection[1], particleDirection[0])
    double vx = v * cos(direction)
    double vy = v * sin(direction)

    x += vx * dt
    y += vy * dt

    return y == yTarget
}
```

Simulaciones



Sistema a simular

- Espacio rectangular LxL.
- N peatones de radio variable, entre rmin y rmax.
- Peatones sin velocidad inicial y con radio inicial rmin.
- Salida de ancho d variable.

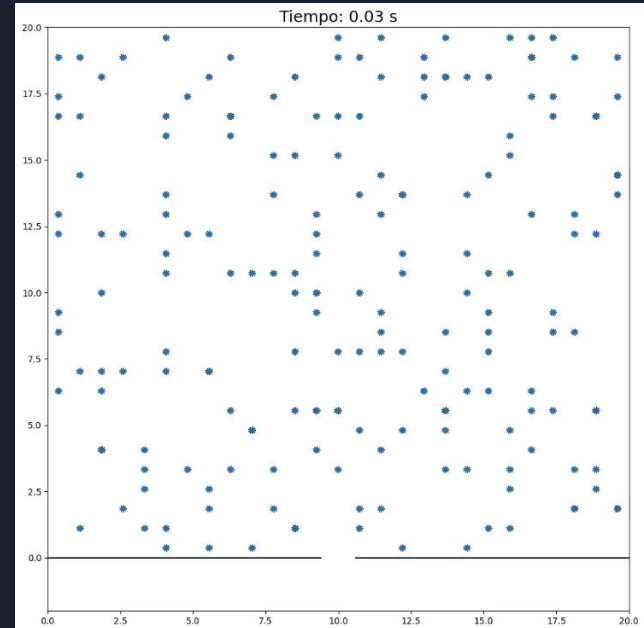
Parámetros fijos:

- L = 20 m
- rmin = 0.10 m
- rmax = 0.37 m
- $\beta = 0.9$
- v_{dmax} = 2 m/s
- $\tau = 0.5$ s
- v_e = v_{dmax}

Parámetros variables:

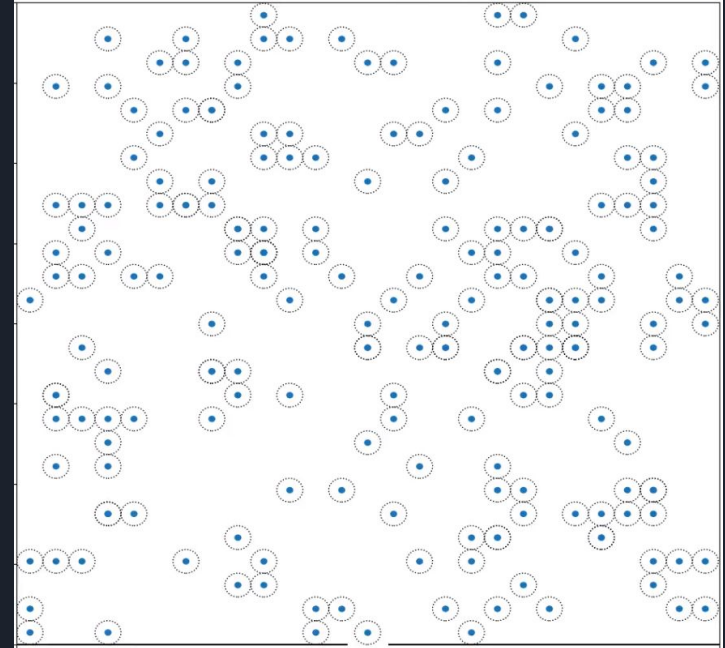
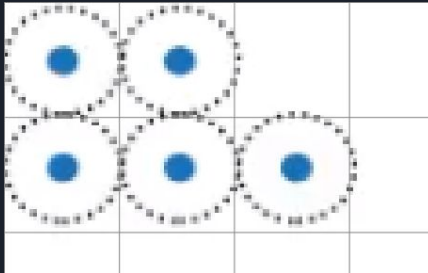
- 200 ≤ N ≤ 380
- 1.2 m ≤ d ≤ 3.0 m

$$\Delta t = \frac{r_{min}}{2v_{dmax}}$$



Peatones en el sistema

- Espacio con grillas de tamaño r_{max}
- Partículas con r_{min} en el centro de cada grilla
- Cada partícula se ubica una grilla libre aleatoria





Estado del sistema

Condiciones iniciales para cada partícula:

- Posición
- Velocidad
- Radio

Se almacenan los nuevos radios, velocidades, y posiciones en cada paso temporal.



Observables

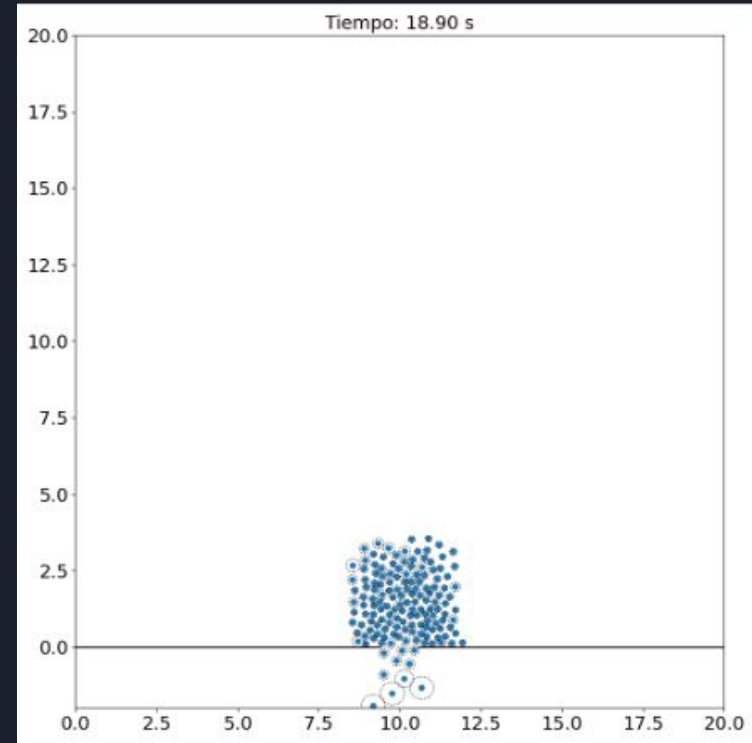
- $Egresos(t)$ = cantidad de personas que salieron del cuarto hasta el tiempo t .
- $Caudal(d)$ = cantidad de personas que salen por segundo, si la salida tiene ancho d .
- Colisiones = cantidad de veces que personas colisionan entre sí y con las paredes.

Resultados



Egreso de personas del cuarto

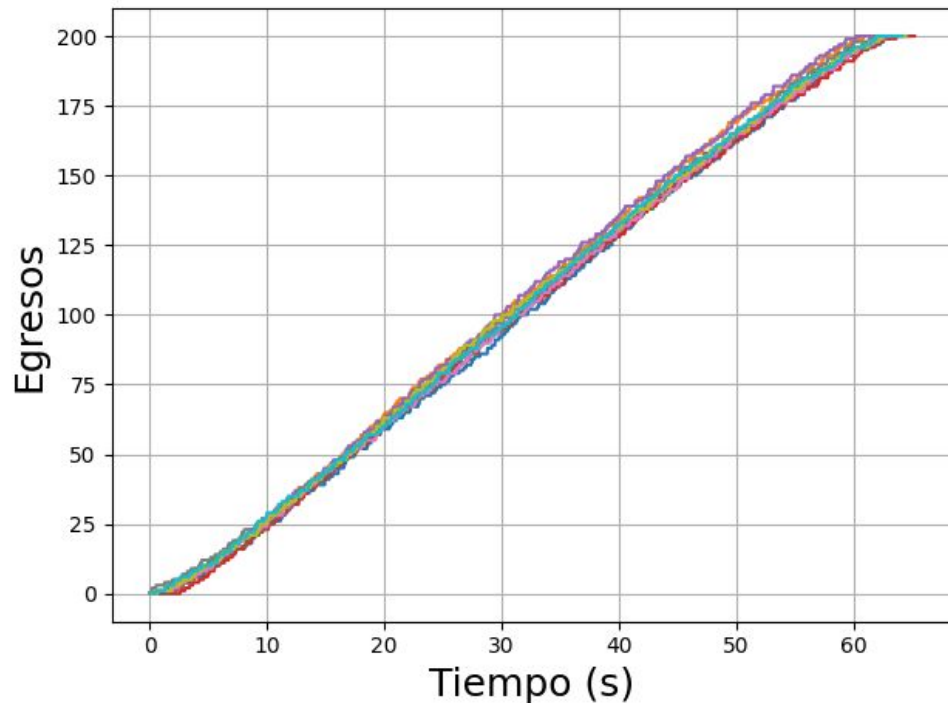
- $N = 200$ personas
- $d = 1.2$ m
- $dt = 0.025$ s
- Visualización cada 5 pasos temporales



https://www.youtube.com/watch?v=T_VxExMDcac

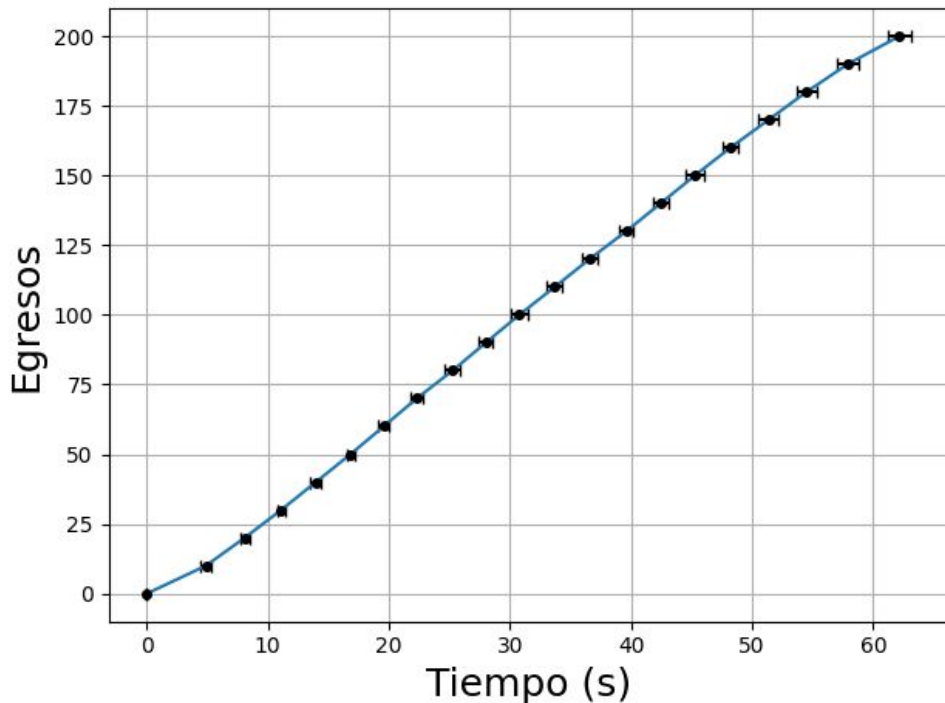
Egreso de personas en el tiempo

- Personas ubicadas en 10 posiciones iniciales distintas.
- $N = 200$ personas
- $d = 1.2$ m
- $dt = 0.025$ s



Promedio de egresos en el tiempo

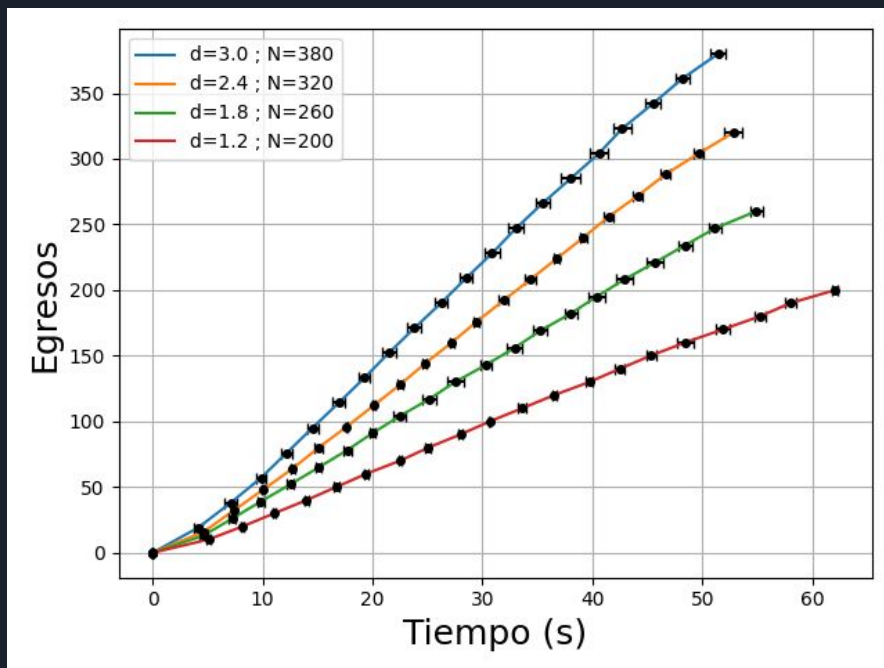
- 10 posiciones iniciales distintas.
- $N = 200$ personas
- $d = 1.2$ m
- $dt = 0.025$ s
- error = desviación estándar



Estado del caudal

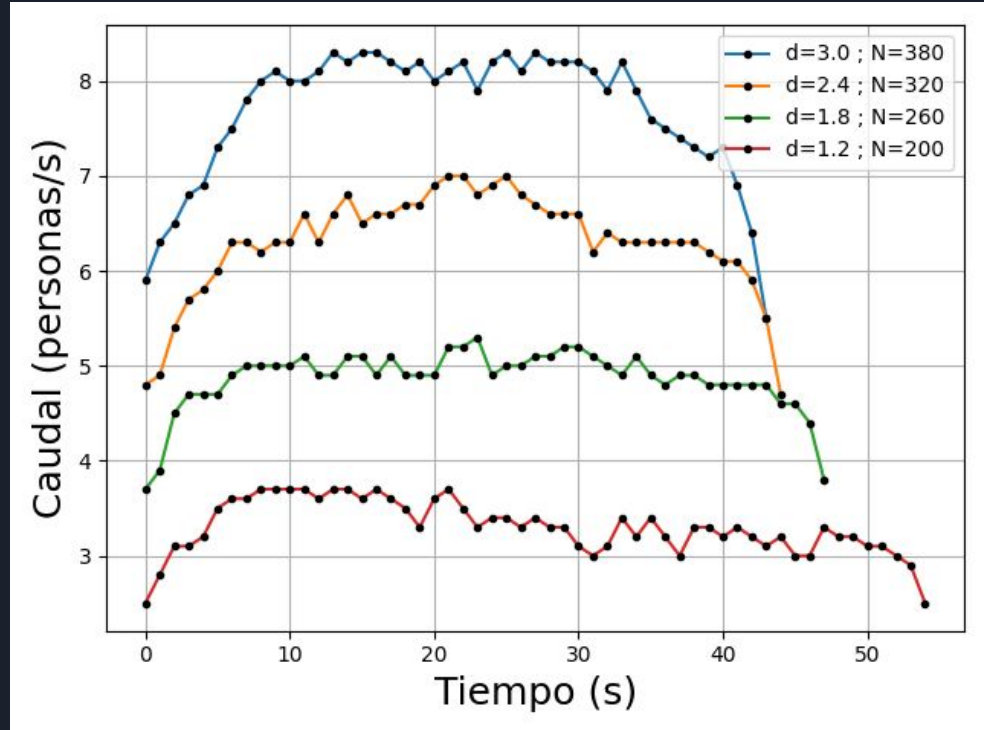
3 realizaciones ; error = desviación estándar

- $[d]$ = metros
- $[N]$ = personas



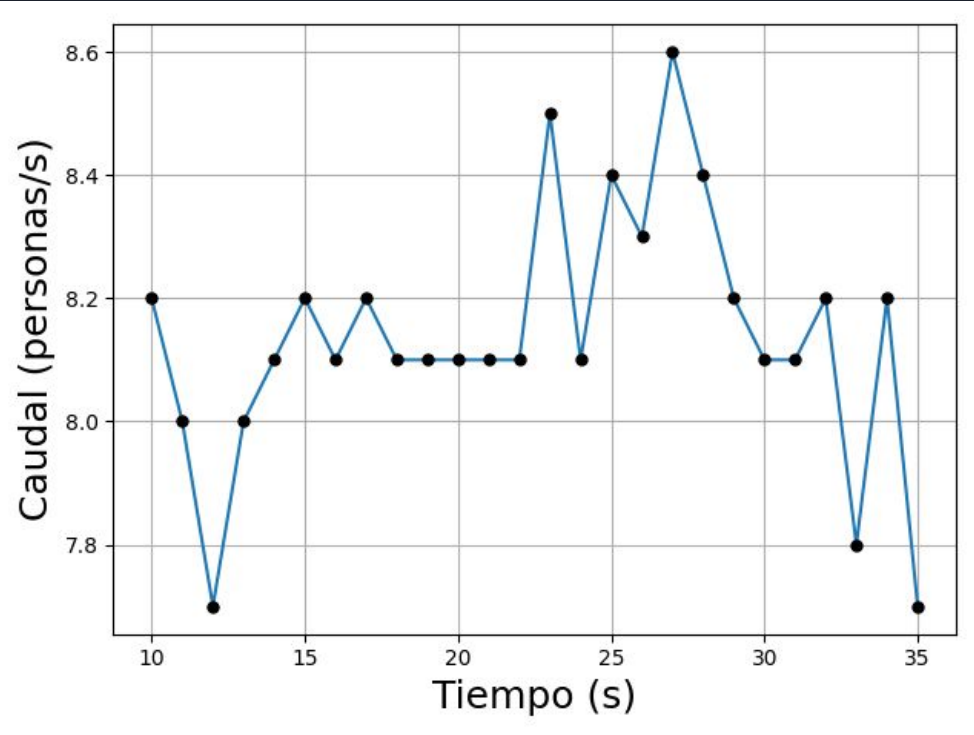
Caudal vs tiempo

- [d] = metros
- [N] = personas
- Ventana de 10s, avanzando cada 1s



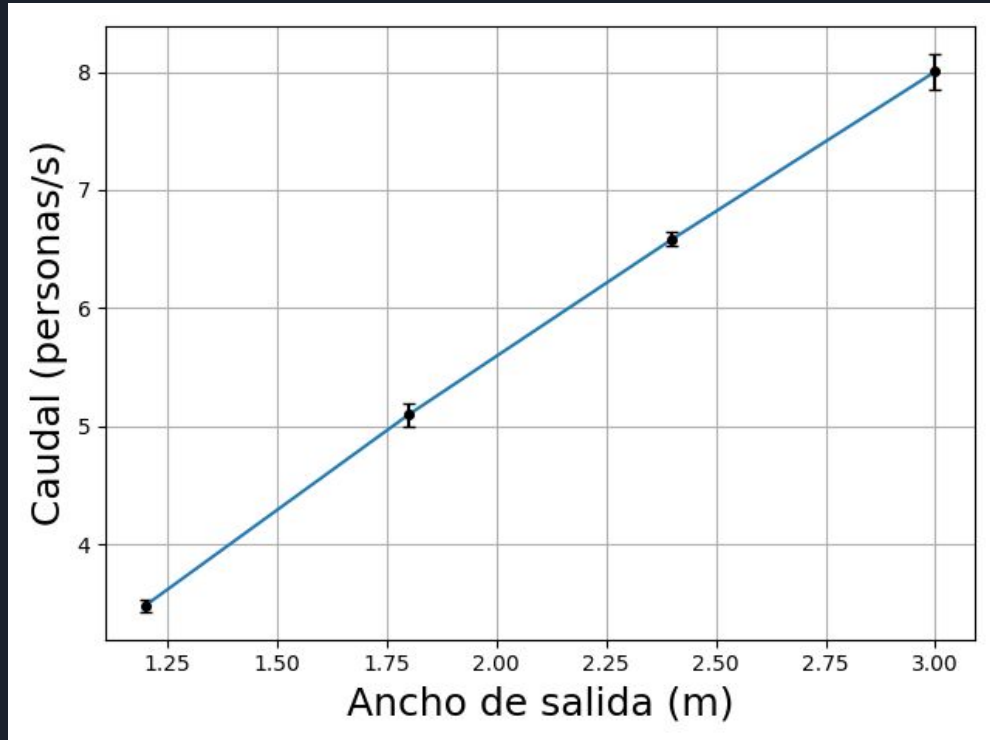
Caudal medio

- $N = 380$ personas
- $d = 3$ m
- Ventana de 10s, avanzando cada 1s



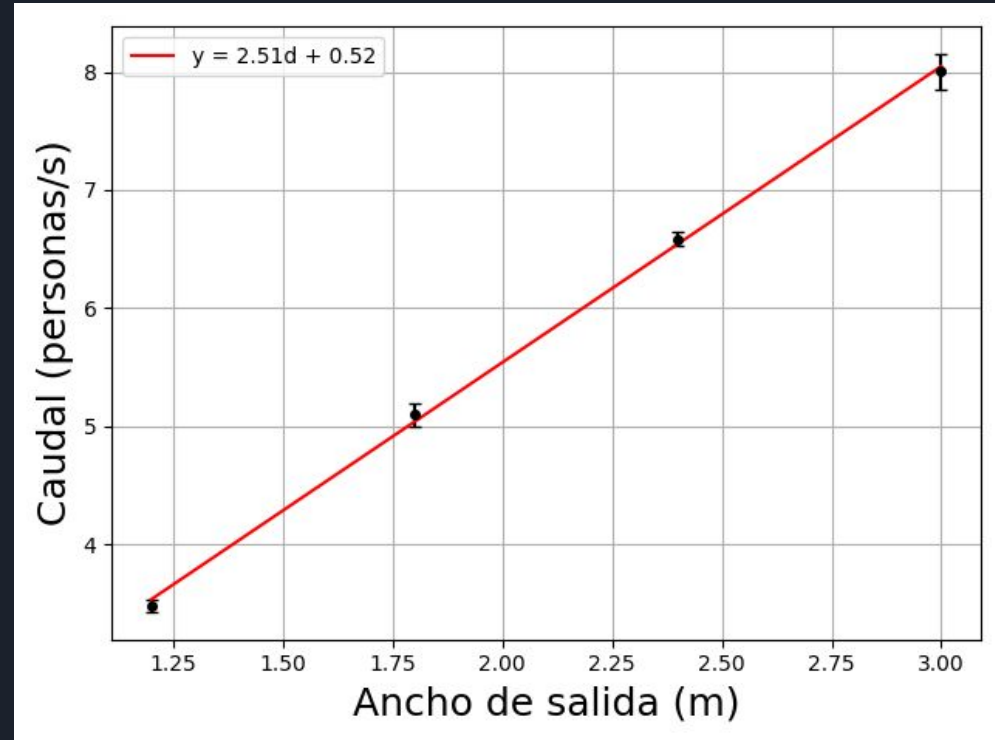
Caudal vs ancho de salida

- 3 realizaciones
- error = desviación estándar



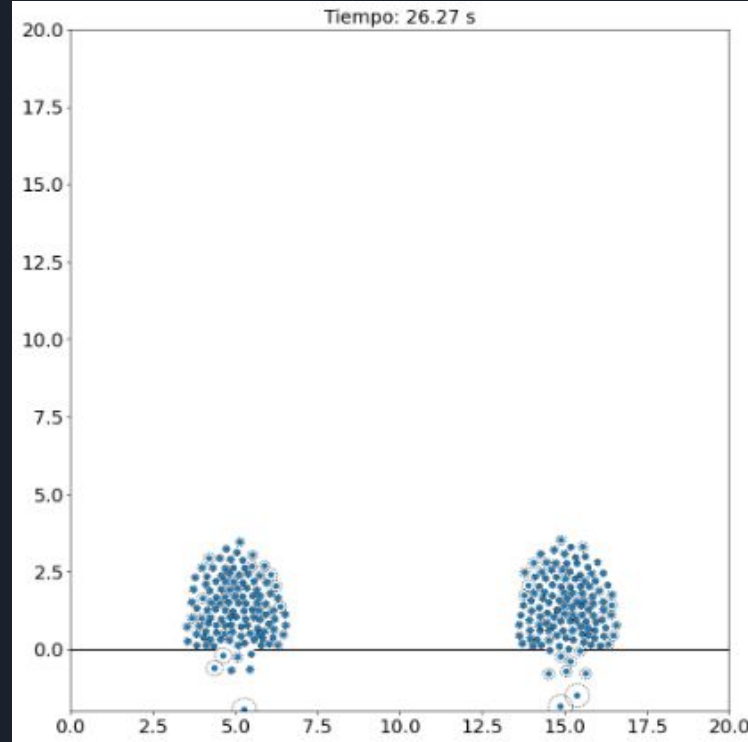
Ajuste lineal del caudal

- 3 realizaciones
- error = desviación estándar



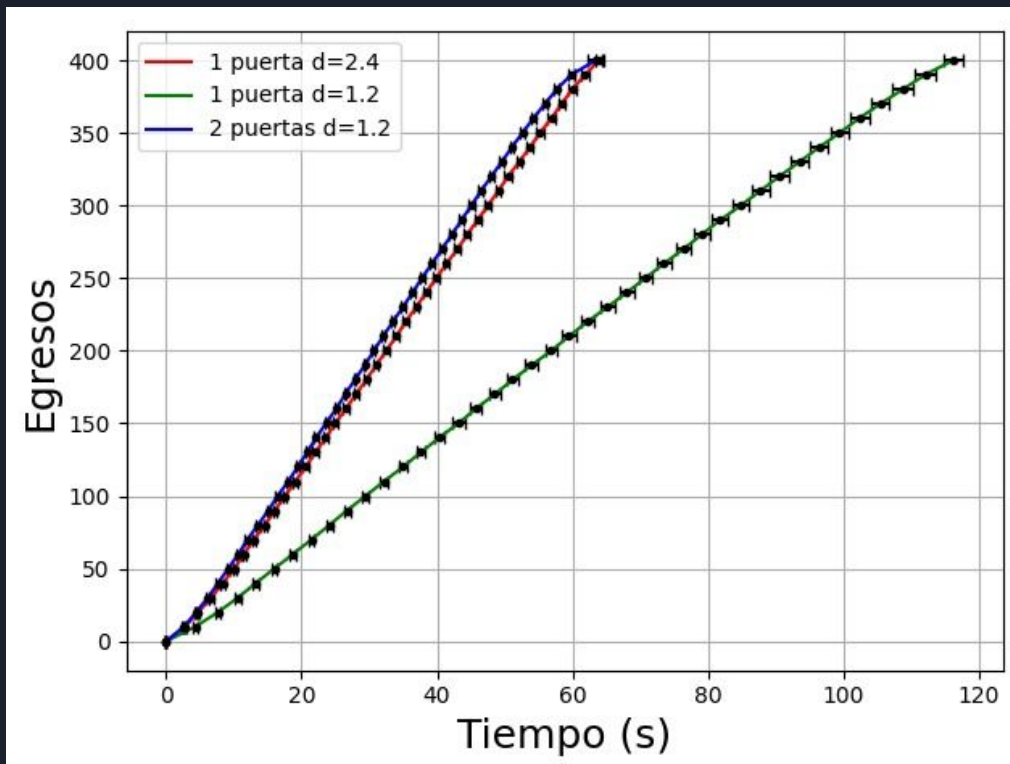
Incorporación de una puerta extra

- $N = 400$ personas
- $d = 1.2$ m
- $dt = 0.025$ s
- Visualización cada 5 pasos temporales



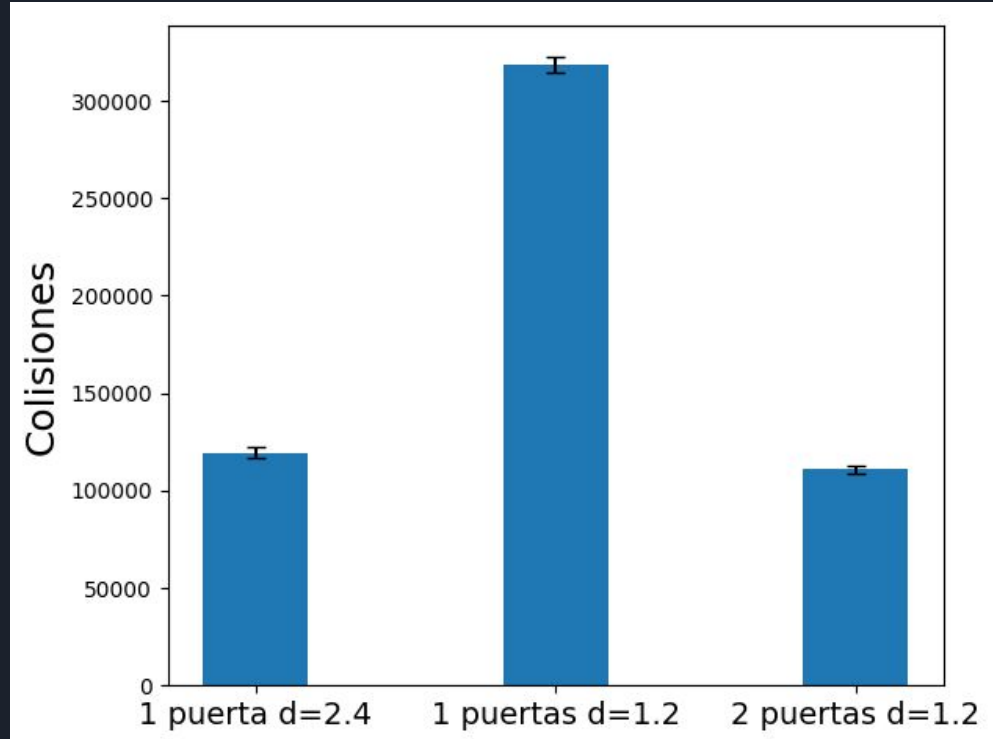
Promedio de egresos en el tiempo

- 10 posiciones iniciales
- $dt = 0.025$ s
- $N = 400$ personas
- error = desviación estándar



Colisiones

- 10 realizaciones
- $N = 400$ personas
- error = desviación estándar



Conclusiones





Conclusiones

- Al acumularse los peatones alrededor de la salida, los egresos crecen linealmente con el tiempo.
- El caudal de peatones a través de la salida es una función lineal de su ancho.
- Incrementar en $\sim 40\text{cm}$ el ancho de la salida, hará que pueda pasar un peatón más por segundo.
- Dos puertas de ancho D mejoran levemente el tiempo de egreso y las colisiones, con respecto a una puerta de ancho $2D$.

Fin

