

**Universidad Autónoma Metropolitana**

**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

---

**Reporte Final  
Simulación de la Dinámica de Peatones**



**CBI | INGENIERÍA EN COMPUTACIÓN**

**SIMULACIÓN DE LA DINÁMICA DE PEATONES**

**ALUMNO | FELIPE DE JESÚS RIOFRÍO LEÓN**

**ASESOR | GERMÁN TÉLLEZ CASTILLO**

## TABLA DE CONTENIDO

<b>TABLA DE CONTENIDO .....</b>	<b>- 1 -</b>
<b>CAPÍTULO I. PROYECTO.....</b>	<b>- 4 -</b>
Antecedentes.....	- 4 -
Objetivos generales .....	- 4 -
Objetivos particulares.....	- 5 -
Justificación .....	- 5 -
<b>CAPÍTULO II. MARCO TEÓRICO .....</b>	<b>- 6 -</b>
Autómatas celulares .....	- 6 -
Descripción de un autómata celular .....	- 7 -
Conceptos .....	- 8 -
<b>CAPÍTULO III. MODELO .....</b>	<b>- 11 -</b>
Modelo de Dinámica de Peatones utilizando AC .....	- 11 -
<b>CAPÍTULO IV. SIMULACIÓN Y RESULTADOS.....</b>	<b>- 15 -</b>
Peor caso.....	- 15 -
Mejor caso .....	- 15 -
Caso promedio .....	- 17 -
<b>CAPÍTULO V. CONCLUSIONES .....</b>	<b>- 18 -</b>
<b>BIBLIOGRAFÍA .....</b>	<b>- 19 -</b>
<b>DOCUMENTOS ANEXOS .....</b>	<b>- 19 -</b>

## CAPÍTULO I. PROYECTO

### Antecedentes

Actualmente, los AC han sido utilizados para modelar fenómenos complejos en diferentes áreas del conocimiento tales como: la dinámica de fluidos, la física estadística, la biología y otros sistemas complejos [1]. Una clase importante de modelos de AC ha sido propuesta para modelar la dinámica del flujo de tráfico [2-3]. Por otra parte, la dinámica peatonal no ha sido estudiada tan extensivamente como el tráfico vehicular; los estudios en este tópico son muy escasos [4-8]. La razón puede ser que el movimiento peatonal es más complejo que el flujo vehicular [9]. Primero, los peatones son más inteligentes que los vehículos y ellos pueden elegir la ruta óptima según el ambiente que los rodea. En segundo lugar, los peatones son más flexibles para cambiar de dirección y no se limitan a “carriles” como en el flujo de vehículos. En tercer lugar, el choque ligero es aceptable y no necesita ser evitado como en los modelos de flujo de tráfico. Un modelo para el movimiento peatonal debe considerar estas diferencias. Actualmente, no se entiende lo suficiente la dinámica peatonal, especialmente para velocidades de caminata diferentes.

El movimiento peatonal es un componente importante en el análisis y diseño en las instalaciones de transporte, pasillos peatonales e intersecciones de tráfico peatonal en mercados y edificios públicos. Una primera aproximación en el modelado de la dinámica de patones es el enfoque de los modelos continuos, los cuales se basan en ecuaciones diferenciales. Dichos modelos tienen el inconveniente de que para ser simulados en una computadora deben de ser discretizados, ocasionando pérdidas de información, lo cual no sucede con los AC los cuales son una herramienta de modelación discreta.

Los AC para la dinámica peatonal han sido propuestos para diferentes escenarios [4-7, 10]. Los modelos en las primeras referencias pueden ser considerados como generalizaciones del Modelo *Bigham-Middleton-Levine* para el tráfico en la ciudad [11], nombrado como modelo de caminata aleatoria parcial.

En este trabajo, es diseñado un modelo de AC que simula la dinámica peatonal que considera el comportamiento humano, en el cual pueden hacerse juicios en situaciones complejas. Este modelo deberá simular el movimiento peatonal con diferentes velocidades de caminata. Dos clases de condiciones de frontera se incluyen para el flujo peatonal, la periódica y la abierta.

### Objetivos generales

Diseñar e implementar un autómata celular (AC) que permita reproducir la dinámica del comportamiento de peatones en pasillos o andenes.

## Objetivos particulares

1. Seleccionar las características que tendrá el AC para resolver el problema planteado en el objetivo general.
2. Diseñar un AC que describa las interacciones de los peatones y el lugar donde éstos se desplazan.
3. Programar el AC diseñado en el objetivo particular dos.
4. Simular el AC programado en el objetivo particular tres.
5. Validar los resultados generados por el objetivo particular cuatro y los publicados en revistas especializadas (ISI).
6. Ajustar el AC diseñado.
7. Crear una interfaz gráfica para el usuario del AC obtenido en el objetivo particular seis.

## Justificación

El movimiento peatonal es un componente importante en el análisis y en el diseño de las instalaciones humanas. Nuestro trabajo pretende examinar el comportamiento de los peatones en andenes o pasillos. Nuestro modelo deberá reproducir comportamiento real de los peatones en un pasillo o andén. Además, el modelo deberá ser capaz de reproducir las diferentes fases que se presentan en la dinámica del comportamiento de peatones, tales como, la fase de movimiento libre, la fase de formación de filas y la fase de parada.

Por otro lado, la computadora juega un papel importante en el estudio y la práctica de la planeación y diseño de supermercados, plazas comerciales, andenes peatonales, etc.

Para el desarrollo de este proyecto terminal, se requiere conocimiento de autómatas y de computación. Un ingeniero en computación puede, a través de prueba y error, inducir y proponer las reglas que permitan al autómata reproducir la dinámica peatonal. Además, el conocimiento de diferentes herramientas de software es necesario para poder seleccionar, la mejor herramienta que permita la implantación del algoritmo usando AC (tópico que se revisa en UEA's como, Inteligencia artificial y Teoría matemática de la computación). Otro de los conocimientos necesarios, es el manejo matemático y computacional del álgebra de matrices; temas cubiertos en UEA's como Álgebra lineal y Complementos de matemáticas; esto es, el conocimiento matemático y computacional requerido para llevar a cabo este proyecto terminal es cubierto por UEA's que pertenecen a la currícula de un ingeniero en computación de la UAM-A.

Se le puede dar continuación a este proyecto mediante el estudio y análisis de la evacuación (ante una situación de sismo, incendio,...) de edificios, escuelas, estadios y cualquier espacio cerrado que concentre población.

## CAPÍTULO II. MARCO TEÓRICO

### Autómatas celulares

Los AC son sistemas dinámicos discretos, creados en la década de 1940's cuando John Von Neumann empezó a considerar la posibilidad de la generación de vida artificial, tratando de que un autómatas fuera capaz de copiarse a sí mismo. Las ideas principales de von Neumann fueron publicadas en el libro *Theory of Self-reproducing Automata*, editado y completado por uno de sus colaboradores A. W. Burks. A sugerencia de su colega Stanislaw Marcim Ulam, Von Neumann utilizó patrones sobre una cuadrícula en el plano, las cuales evolucionan según una regla de transformación fija. De esta forma el problema de auto reproducción mecánica, quedaba reducido a la búsqueda de ciertas configuraciones que, con la aplicación de la regla, dieran lugar a copias idénticas.

La historia de los AC puede ser clasificada en tres etapas asociadas a los nombres de los científicos que en cada momento marcaron un punto de inflexión en el desarrollo de la teoría: la era de Von Neumann, la era de Martin Gardner y la era de Stephen Wolfram.

**Era de Von Neumann.** La primera etapa la inicia von Neumann [12], quien una vez terminada su participación en el desarrollo de la primera computadora ENIAC tenía en mente desarrollar una máquina con la capacidad de construir a partir de sí misma otras máquinas (auto-reproducción) y soportar comportamiento complejo. Con la ayuda de su amigo Stanislaw Ulam, von Neumann implementa sus ideas y conceptos en un vector de dos dimensiones  $\mathbf{Z} \times \mathbf{Z}$  (donde  $\mathbf{Z}$  representa el conjunto de los enteros). El vector es llamado el espacio de evoluciones y cada una de las posiciones (llamadas células) en el vector toma un valor de conjunto finito de estados  $|\mathbf{k}| = 29$ . La función de transición que determina el comportamiento del AC utiliza una vecindad, hoy llamada, vecindad de von Neumann; esta vecindad consiste en un elemento central  $\mathbf{x}(\mathbf{i}, \mathbf{j})$  (llamada célula central) y sus vecinos que son las células  $\mathbf{x}(\mathbf{i}, \mathbf{j}-1)$ ,  $\mathbf{x}(\mathbf{i}, \mathbf{j}+1)$ ,  $\mathbf{x}(\mathbf{i}-1, \mathbf{j})$  y  $\mathbf{x}(\mathbf{i}+1, \mathbf{j})$  (es decir, la célula en cuestión y sus células vecinas más próximas, arriba, abajo, izquierda y derecha, respectivamente).

**Era de Martin Gardner.** En 1970, John Horton Conway dio a conocer el AC que probablemente sea el más conocido: "El Juego de la vida" (*Life*), publicado por Martin Gardner en su columna *Mathematical Games* en la revista *Scientific American* [13]. *Life* ocupa una cuadrícula (lattice bidimensional) donde se coloca al inicio un patrón de células "vivas" o "muertas". La vecindad para cada célula son los ocho vecinos formados por la vecindad de Von Neumann y las cuatro células de las dos diagonales (esta vecindad hoy se conoce como vecindad de Moore). De manera repetida, se aplican simultáneamente sobre todas las células de la cuadrícula las siguientes 3 reglas:

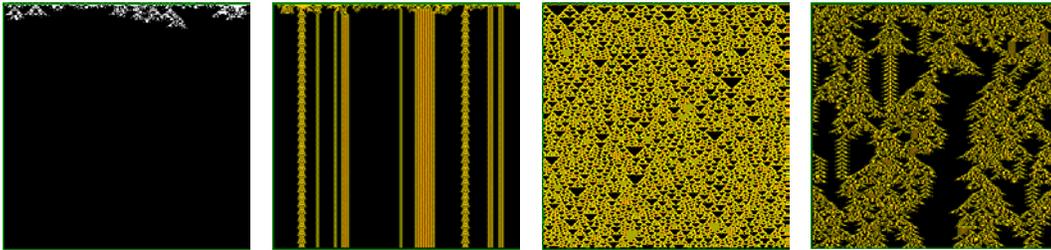
1. **Nacimiento:** se reemplaza una célula muerta por una viva si dicha célula tiene exactamente 3 vecinos vivos.
2. **Muerte:** se reemplaza una célula viva por una muerta si dicha célula no tiene más de 1 vecino vivo (muerte por aislamiento) o si tiene más de 3 vecinos vivos (muerte por sobrepoblación).
3. **Supervivencia:** una célula viva permanecerá en ese estado si tiene 2 o 3 vecinos vivos.

Una de las características de *Life* es su capacidad de realizar cómputo universal, es decir, que con una distribución inicial apropiada de células vivas y muertas, *Life* se puede convertir en una computadora de propósito general (máquina de Turing).

**Era de Stephen Wolfram.** Stephen Wolfram [14] ha realizado numerosas investigaciones sobre el comportamiento cualitativo de los AC. Con base en su trabajo sobre AC unidimensionales, con dos o tres estados, sobre configuraciones periódicas que se presentan en el AC, observó sus evoluciones para configuraciones iniciales aleatorias. Así, dada una regla, el AC exhibe diferentes comportamientos para diferentes condiciones iniciales.

Wolfram clasificó en cuatro clases el comportamiento cualitativo de los AC unidimensionales. (ver *figura 1*):

- **Clase I.** La evolución lleva a una configuración estable y homogénea, es decir, todas las células terminan por llegar al mismo valor.
- **Clase II.** La evolución lleva a un conjunto de estructuras simples que son estables o periódicas.
- **Clase III.** La evolución lleva a un patrón caótico.
- **Clase IV.** La evolución lleva a estructuras aisladas que muestran un comportamiento complejo (es decir, ni completamente caótico, ni completamente ordenado, sino en la línea entre uno y otro, este suele ser el tipo de comportamiento más interesante que un sistema dinámico puede presentar).



**Figura 1.** Ejemplos de las clases de los AC según Stephen Wolfram. De izquierda a derecha se observan las clasificaciones I a IV.

Los AC pueden ser usados para modelar numerosos sistemas físicos que se caractericen por un gran número de componentes homogéneos que interactúen localmente entre sí; de hecho, cualquier sistema real al que se le puedan analogar los conceptos de "vecindad", "estados de los componentes" y "función de transición" es candidato para ser modelado por un AC. Algunos ejemplos de áreas en donde se utilizan los AC son:

- Modelación de flujo de tránsito vehicular y de peatonal.
- Modelación de fluidos (gases o líquidos).
- Modelación de la evolución de células o virus como el VIH.
- Modelación de procesos de percolación.
- Etc.

## Descripción de un autómeta celular

Un AC consta de los siguientes elementos:

- Un enrejado, o retícula, o arreglo ó lattice regular de celdas o células o sitios.
- A cada celda se le asigna un valor, llamado estado; donde el conjunto de estados es finito.
- Cada celda tiene un conjunto finito de celdas vecinas, llamada vecindad de la célula. La relación de vecindad es uniforme.

# Dinámica de Peatones

- Una función de transición que es discreta y que indica cuál será el estado de una celda en el instante  $t+1$ , con base en su estado y los estados de su vecindad en el tiempo  $t$ . La función es la misma para todas las celdas y para todo el tiempo.

## Conceptos

- **Definición 1.** Una retícula (o una lattice), es un arreglo regular de elementos de dimensión  $d \in \mathbb{Z}^+$ ; los elementos del lattice son llamados celdas o células. A cada célula se le asigna un valor de una familia de conjuntos finitos llamados conjuntos de estados (ver *figura 2*).
- **Definición 2.** Una lattice  $L$ , es homogénea si todas sus celdas toman su valor a partir del mismo conjunto  $S$ .
- **Definición 3.** Sea  $L$  una lattice homogénea y  $r \in L$  una célula, el estado de  $r$  es un elemento  $s \in S$ , esto es  $r \leftarrow s$ .

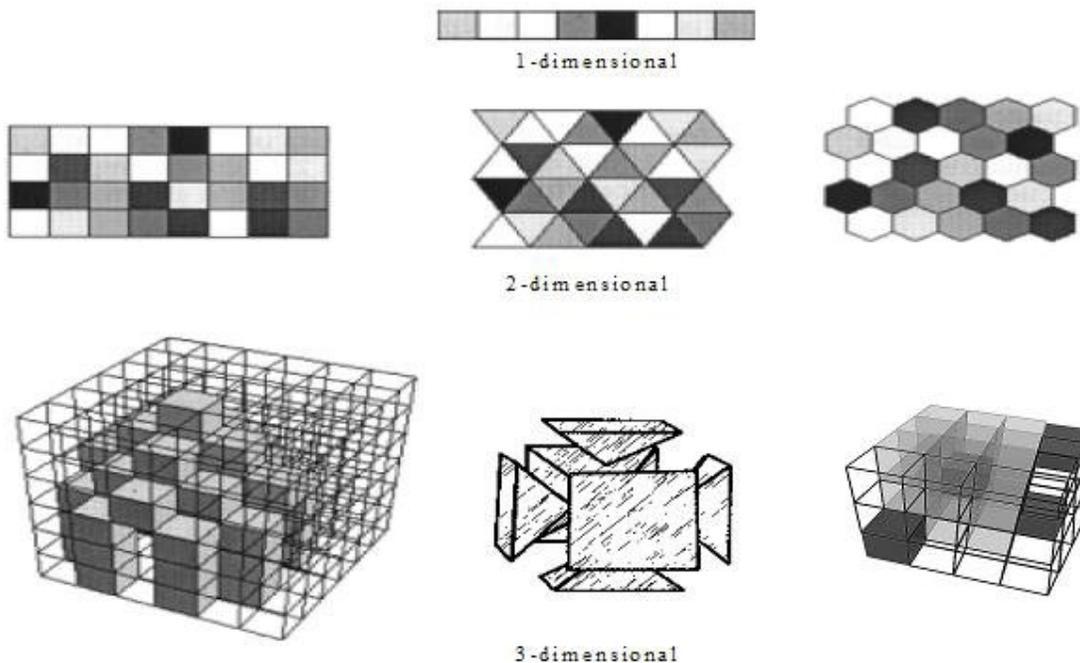


Figura 2. Lattices de una, dos y tres dimensiones, note que las células no necesariamente son cuadradas.

El tiempo avanza en etapas discretas y la dinámica está dada por una regla explícita llamada función local. La función local es usada en cada etapa de tiempo sobre cada célula para determinar su nuevo estado a partir del estado actual de ciertas células, llamadas la vecindad de la célula, es decir, la función local toma los estados de la vecindad de la célula como argumento y devuelve como resultado el nuevo estado de la célula correspondiente. Las células cambian sus estados en etapas de tiempo discreto de acuerdo a la función local. Estos cambios de estado de las células podrían o no realizarse de forma sincronizada para todas las células del lattice, dependiendo de la naturaleza del problema que se modele. Si la lattice es homogénea todas las células operan bajo la misma función local.

## Dinámica de Peatones

*Observación:* La **definición 1** implica que una lattice puede ser infinita, sin embargo, implementar esto en un programa de computadora es imposible, por esta razón, las lattices se consideran compuestas por un número finito de células, y se implementan siguiendo algunas de las siguientes condiciones de frontera:

- **Frontera periódica.** Es una frontera en donde los bordes opuestos de una lattice se consideran conectados. Para una lattice de una dimensión, la frontera periódica permite visualizar geoméricamente al lattice en dos dimensiones como una circunferencia. Para una lattice de dos dimensiones, esta frontera permite visualizar geoméricamente al lattice en tres dimensiones como un “toroide” (ver *figura 3*).

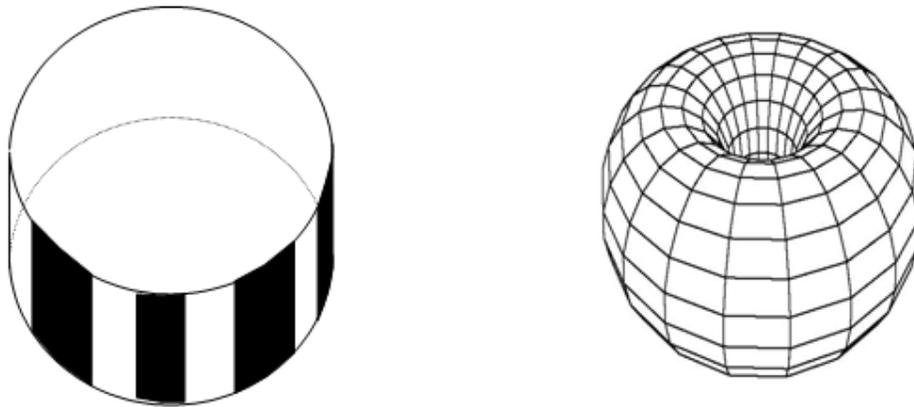


Figura 3. Lattice con frontera periódica de varias dimensiones.

- **Frontera abierta.** Es una frontera en donde fuera de ella, en la lattice se consideran que existen células con un valor fijo tomado de  $S$ .
- **Frontera reflectora.** Es una frontera en donde los valores de las células en la frontera de la lattice se consideran reflejados fuera de esta.
- **Sin frontera.** Es una lattice que comienza con algún tamaño finito y que crece dinámicamente conforme se va requiriendo.  
*Nota:* No debe confundirse con una definición anterior que implica una lattice inicialmente infinita.
- **Combinación de las anteriores.** En algunos casos podemos considerar los extremos de la lattice con alguno de los tipos de frontera anteriormente citados, o que en otro extremo podría tener alguna otra condición. Esto le dará a la lattice una geometría específica, por ejemplo si para una lattice en dos dimensiones se maneja una frontera periódica en dos extremos opuestos y una frontera abierta en los otros dos extremos, la lattice puede visualizarse en tres dimensiones como una banda circular.
- En lo sucesivo, dada una lattice  $L$ , esta será considerada como una lattice homogénea finita con cierta condición de frontera; además se identificará al índice  $j$  con la  **$j$ -ésima** célula de  $L$ , con  $j \in \{0, \dots, n-1\} \in \mathbb{Z}$ , siendo  $n$  el número de células de  $L$ .

Otro concepto esencial en la definición de un AC es el de vecindad; a cada célula de una lattice le corresponde un conjunto de células de la misma lattice:

- **Definición 4.** Sea  $L$  una lattice y sea  $r \in L$  una célula.  
Una vecindad de tamaño  $n \in \mathbb{Z}^+$  para  $r$ , es un conjunto finito de células  $N(r) = \{k_1, \dots, k_n\} \subset L$  tal que:  
 $r = k_j$  para algún  $j$ , si  $r \in N(r)$  o para ningún  $j$  en caso contrario.
- **Definición 5.** Un AC es una tupla  $(L, S, N, f)$  donde:
  1.  $L$  es una lattice de dimensión  $d$ , con  $d \in \mathbb{Z}^+$ . En el caso de una lattice finita, esta consiste de  $k$  células, y se le implementa con cierta condición de frontera.
  2.  $S = \{0, 1, 2, \dots, k-1\}$  es un conjunto finito de estados.
  3.  $N$  es un conjunto de vecindades definido por:  
 $N = \{N(r) : r \in L \text{ es una célula y } N(r) \text{ es la vecindad de } r \text{ de tamaño } n \in \mathbb{Z}^+ \forall r \in L\}$
  4.  $f: S^n \rightarrow S$  es una función llamada función de transición o función local. Esta función se aplica en cada paso de tiempo discreto sobre las células de  $L$ , tomando como argumentos los estados de las células de  $N(r)$ , y regresando como resultado el nuevo valor de  $r$  para el siguiente paso de tiempo.
- **Definición 6.** Un AC  $A = (L, S, N, f)$  se dice que es homogéneo si y solamente si:
  1.  $L$  es homogénea.
  2.  $f$  se aplica a cada paso discreto de tiempo sobre todas las células de  $L$  por igual.

Si al menos se cumple 1, se dice que  $A$  es un AC de lattice homogénea. Al conjunto de estados de todas las células del AC se le conoce como configuración del AC:

- **Definición 7.** Sea  $A = (L, S, N, f)$  un AC. Una configuración de  $A$  es una función  $T_t: L \rightarrow S$  que asocia a cada célula de la lattice  $L$  en el tiempo  $t$ , un estado de  $S$ . Dada una configuración, asignarle nuevos estados a todas las células para generar una nueva configuración, a través de la función de transición del AC, se conoce como función global.
- **Definición 8.** Sea  $A = (L, S, N, f)$  un AC. La función global es una función  $F: L \rightarrow L$  que se aplica evaluando  $F$  sobre cada una de las células del conjunto de células del AC en el tiempo  $t$ , es decir, sobre una configuración  $T_t$  del AC en el tiempo  $t$ , y regresa nuevos valores para todas las células del AC, es decir, se pasa de la configuración  $T_t$  de  $A$ , a la configuración  $T_{t+1}$  de  $A$ , del tiempo  $t$  al  $t+1$ .

## CAPÍTULO III. MODELO

### Modelo de Dinámica de Peatones utilizando AC

Para poder modelar el flujo de peatones en un pasillo es necesario definir lo siguiente:

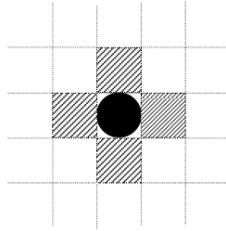
**Definición 9.** Un modelo de dinámica de peatones es una colección de partículas, que representan a los peatones moviéndose sobre una línea, ésta línea representa la vía sobre la que circulan los mismos. Esos peatones se mueven (circulan) de acuerdo a consideraciones de cada uno dependiendo la situación en la que se encuentren. La velocidad, dependerá de las condiciones del recinto, así como de la densidad de peatones.

A continuación se describe el modelo propuesto para la Dinámica de Peatones:

- El sistema utilizado para este modelo es de **dos** dimensiones.
- La lattice es de tamaño **m x n**, lo cual permitirá un mejor análisis en el modelado.
- Cada célula (cada sitio de la lattice) puede estar:
  1. **Vacía** u,
  2. **Ocupada** por: un muro, una pared (obstáculo) o por un peatón.
- El modelo incluye movimiento peatonal con **cuatro** diferentes velocidades de caminata:
  1. **Dos** que van a **1m/s**, pero en direcciones opuestas y
  2. **Dos** que van **1.5 m/s.**, pero en direcciones opuestas.
- Existen además, **dos** clases de peatones que denominaremos:
  1. **Peatón Derecho**, que llamaremos **Caminante Derecho (CD)** y
  2. **Peatón Izquierdo**, que llamaremos **Caminante Izquierdo (CI)**.
- Los **CD** van con velocidades de caminata de **1.0** y **1.5 m/s** y se mueven desde el **límite izquierdo** al **límite derecho**.
- Los **CI** van con velocidades de caminata de **1.0** y **1.5 m/s** y se desplazan desde el **límite derecho** al **límite izquierdo**.

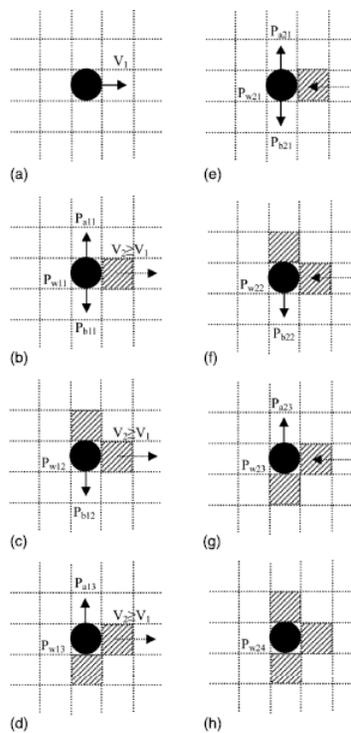
# Dinámica de Peatones

- La vecindad a ocupar en este modelo es la de Von Neumann (ver *figura 4*).



**Figura 4.** Vecindad Von Neumann.

- En la *figura 5* se muestran algunos de los casos posibles de situaciones de iteración entre peatones:



**Figura 5.** Casos de iteración de peatones en un pasillo (a)-(h).

- Para los peatones con velocidades de caminata de:
  1. **1.0 m/s** el estado de la célula se actualiza en cada **3** incrementos de tiempo (o pasos de tiempo).
  2. **1.5 m/s** el estado se actualiza cada **2** incrementos de tiempo
- El tamaño de una célula corresponde aproximadamente a:  $0.5 \times 0.5 \text{ m}^2$
- Para este modelado se utiliza la siguiente notación:
  1.  $P_a$ , significa la probabilidad de que el peatón vaya a la célula vecina de arriba.

## Dinámica de Peatones

2.  $P_b$ , significa la probabilidad de que el peatón vaya a la célula vecina de abajo.
3.  $P_w$ , significa la probabilidad de que el peatón se espere.

- ❖ En la *figura 5(a)*, la célula adyacente derecha está desocupada. El peatón se moverá a la derecha independiente del estado anterior. Pero si se ocupa la célula adyacente derecha, la opción de su ruta depende del estado anterior.
- ❖ En las *figuras 5(b) - 5(d)*, la célula adyacente derecha es ocupada por el peatón con velocidad mayor o igual de caminata a la suya.
- ❖ En las *figura 5(e) - 5(g)*, la célula adyacente derecha es ocupada por el peatón izquierdo.
- ❖ Para los casos (*e - g*), nosotros consideramos la siguiente regla de tráfico:

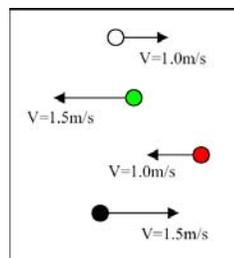
### ***Obligar al peatón a caminar por el lado derecho de su ruta.***

- ❖ El peatón tiene la prioridad de moverse en el lado derecho de su ruta; por lo que, las probabilidades de la transición de peatones son los movimientos preferentemente en la dirección derecha en las configuraciones (*e - g*), es decir:

1.  $P_{b21} > P_{a21}$  y
2.  $P_{b22} > P_{a23}$

- ❖ Para el caso (*h*), si las células adyacentes (arriba, abajo y a la derecha) están ocupadas, el peatón tendrá que parar y esperar. Si el peatón tiene velocidad de caminata de 1.5 m/s, entonces rebasará al peatón que tiene una velocidad de caminata más baja.

- En este modelo, la actualización es secuencial y se toma una decisión sobre la célula a ocupar de manera estocástica.
- En cada paso-tiempo de actualización, los peatones están numerados aleatoriamente del **1** a **N**, donde **N** es el número total de peatones en el sistema, y cada peatón es actualizado una vez en orden secuencial (ie., del **1** a **N**).
- El la *figura 6* se muestra la ilustración esquemática del flujo peatonal en el sistema, que se compone de paredes inferior y superior. Los peatones que van a la derecha con velocidad de caminata de 1.0 y 1.5 m/s son indicadas por círculos blanco y negros, respectivamente, y los peatones izquierdos que van con velocidad de caminata de 1.0 y 1.5 m/s son indicadas por círculos rojo y verde.



**Figura 6.** Flujo de los dos tipos de peatones.

## Dinámica de Peatones

---

---

- En este proyecto consideramos dos condiciones de frontera:
  1. **Periódica** y
  2. **Abierta**.
- Para las **condiciones de frontera periódicas**, los dos tipos de peatones (**CD** y **CI**) se distribuyen en forma aleatoria inicialmente:
  1. Si el peatón llega al límite derecho, él se traslada al límite izquierdo, y
  2. Si el peatón izquierdo llega al límite izquierdo, él se traslada al límite derecho.

ie., el número total de peatones de cada tipo se conserva:

**Sean:**  $\mathbf{p_{plr} W^2}$ ,  $\mathbf{p_{phr} W^2}$ ,  $\mathbf{p_{pll} W^2}$ , y  $\mathbf{p_{phl} W^2}$ ,

**donde:**  $\mathbf{p_{plr}}$ ,  $\mathbf{p_{phr}}$ ,  $\mathbf{p_{pll}}$ , y  $\mathbf{p_{phl}}$  son las densidades iniciales para los peatones blanco, negro, rojo y verde, respectivamente, entonces, la densidad total es:

$$\mathbf{p_p = p_{plr} + p_{phr} + p_{pll} + p_{phl}} \quad \text{y}$$
$$\mathbf{p_{plr} = p_{phr} = p_{pll} = p_{phl}}$$

- Para las **condiciones de frontera abierta**:
  1. La densidad de peatones con velocidades de caminata de **1.0 m/s** es  $\mathbf{p_{olr} W}$ , y para los de **1.5 m/s** es  $\mathbf{p_{ohr} W}$  y son distribuidas aleatoriamente en el límite izquierdo.
  2. Similarmente, las densidades de peatones izquierdos son:  $\mathbf{p_{oll} W}$  y el  $\mathbf{p_{ohl} W}$  y se distribuyen en forma aleatoria en el límite derecho.
  3. Si los peatones llegan al límite derecho y los peatones izquierdos llegan al límite izquierdo, ellos son borrados del sistema.
  4. En este trabajo, solamente se consideran los casos:

$$\mathbf{p_{olr} = p_{ohr} = p_{oll} = p_{ohl} = p_o}$$

**donde:**  $\mathbf{p_{olr}}$ ,  $\mathbf{p_{oll}}$ ,  $\mathbf{p_{ohr}}$ , y  $\mathbf{p_{ohl}}$  son las densidades de entrada para los peatones blanco, negro, rojo y verde, y  $\mathbf{p_o}$  es la densidad de la entrada de peatones de cada límite.

## CAPÍTULO IV. SIMULACIÓN Y RESULTADOS

Se realizaron diversas simulaciones con los mismos parámetros de entrada, variando únicamente el porcentaje total de peatones existentes en el sistema y deteniendo la simulación en la iteración número cien. Dentro de las diferentes simulaciones realizadas, se presentan en este reporte tres simulaciones para ejemplificar los casos: peor caso, caso promedio y mejor caso.

### Peor caso

Dentro de este caso se utilizó un porcentaje de peatones igual a sesenta (60 %). La simulación comenzó con un tránsito de peatones congestionado pero como era de esperarse a medida que transcurrieron más iteraciones de la simulación, comenzó a haber un congestionamiento mayor dentro del pasillo a tal grado de formarse un bloqueo en ambas direcciones. La *figura 7* muestra una captura de la pantalla principal de la simulación que contiene los parámetros con los cuales se produjo este caso así como su respectiva representación en la gráfica.

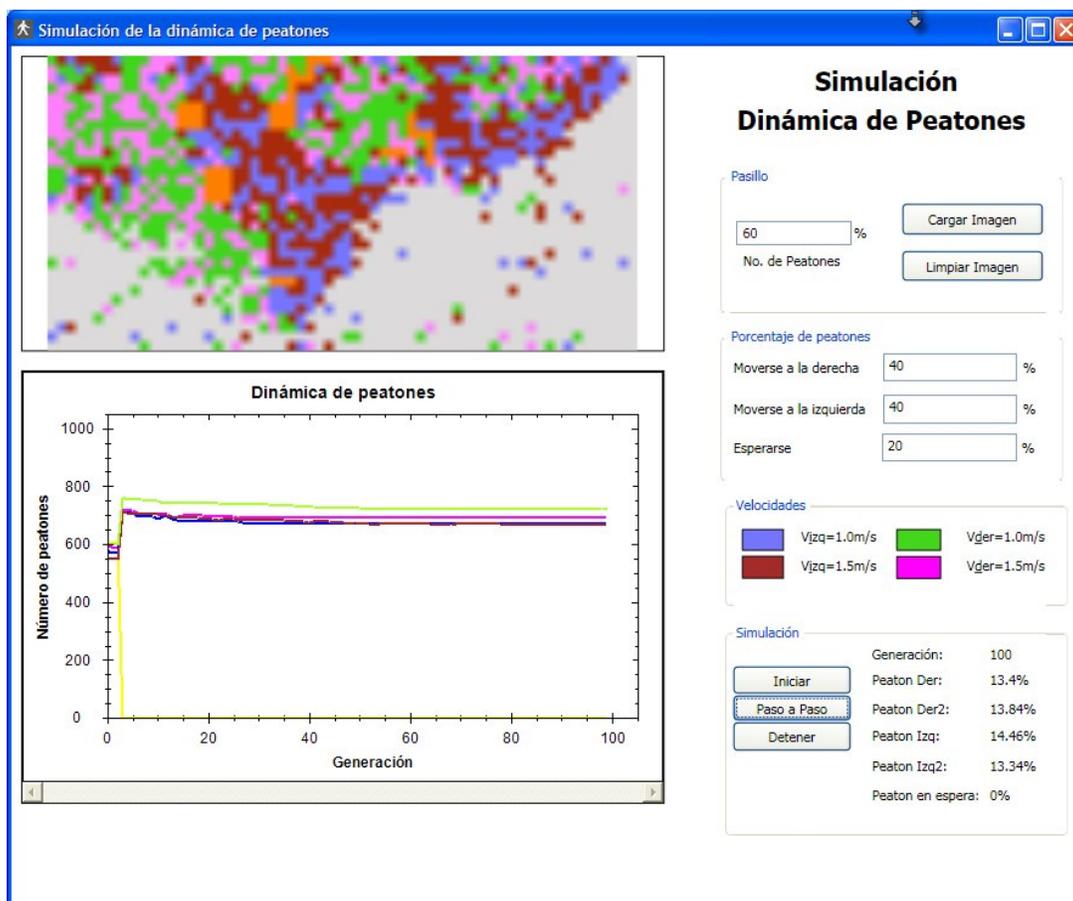


Figura 7. Pantalla que representa el Peor Caso.

## Caso promedio

Dentro de este caso se utilizó un porcentaje de peatones igual a treinta (30 %). La simulación comenzó con un tránsito de peatones moderado pero como era de esperarse a medida que transcurrieron más iteraciones de la simulación, comenzó a formarse algún tipo de bloqueo pero que aun así permitía el flujo de los peatones en ambas direcciones. La figura 8 muestra una captura de la pantalla principal de la simulación que contiene los parámetros con los cuales se produjo este caso así como su respectiva representación en la gráfica.

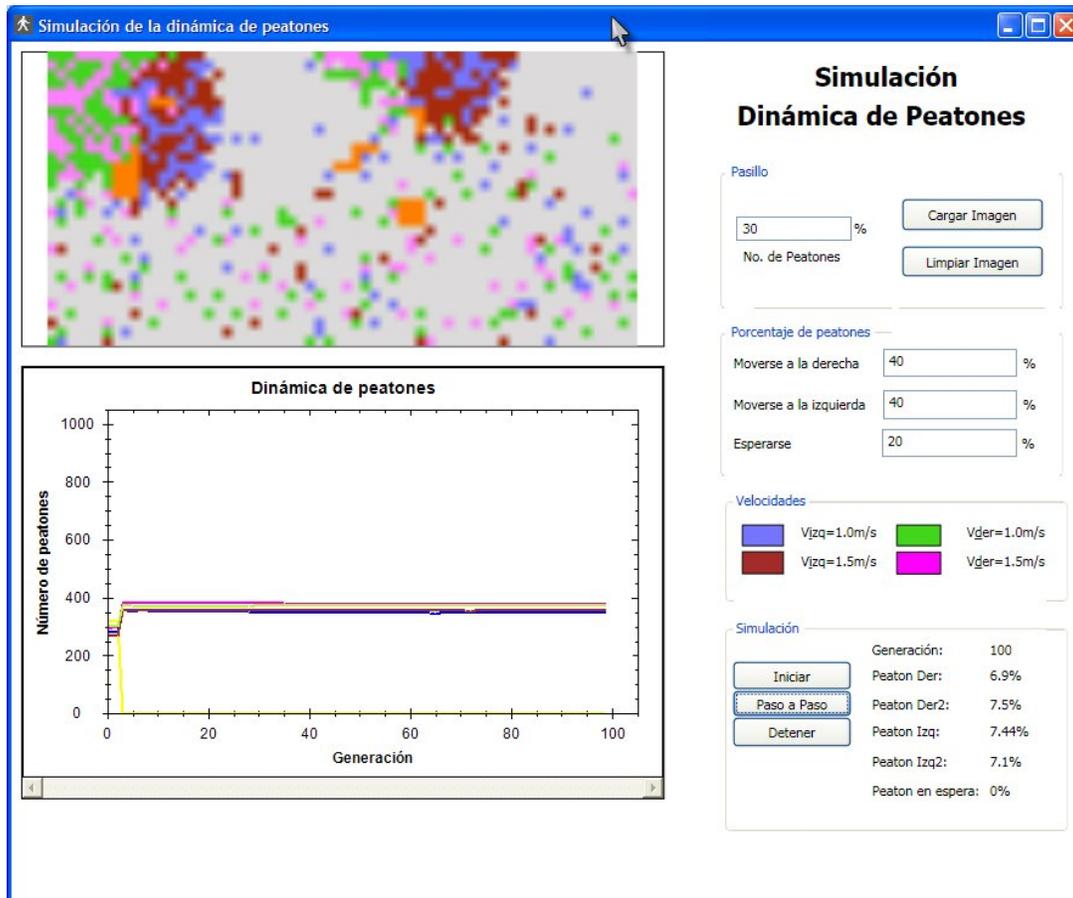


Figura 8. Pantalla que representa el Caso Promedio.

## Mejor caso

Dentro de este caso se utilizó un porcentaje de peatones igual a diez (30 %). La simulación comenzó con un tránsito de peatones bajo pero nunca dejó de haber presencia de ellos dentro del pasillo, esto permite que no se forme ningún bloqueo en ambas direcciones. La figura 9 muestra una captura de la pantalla principal de la simulación que contiene los parámetros con los cuales se produjo este caso así como su respectiva representación en la gráfica.

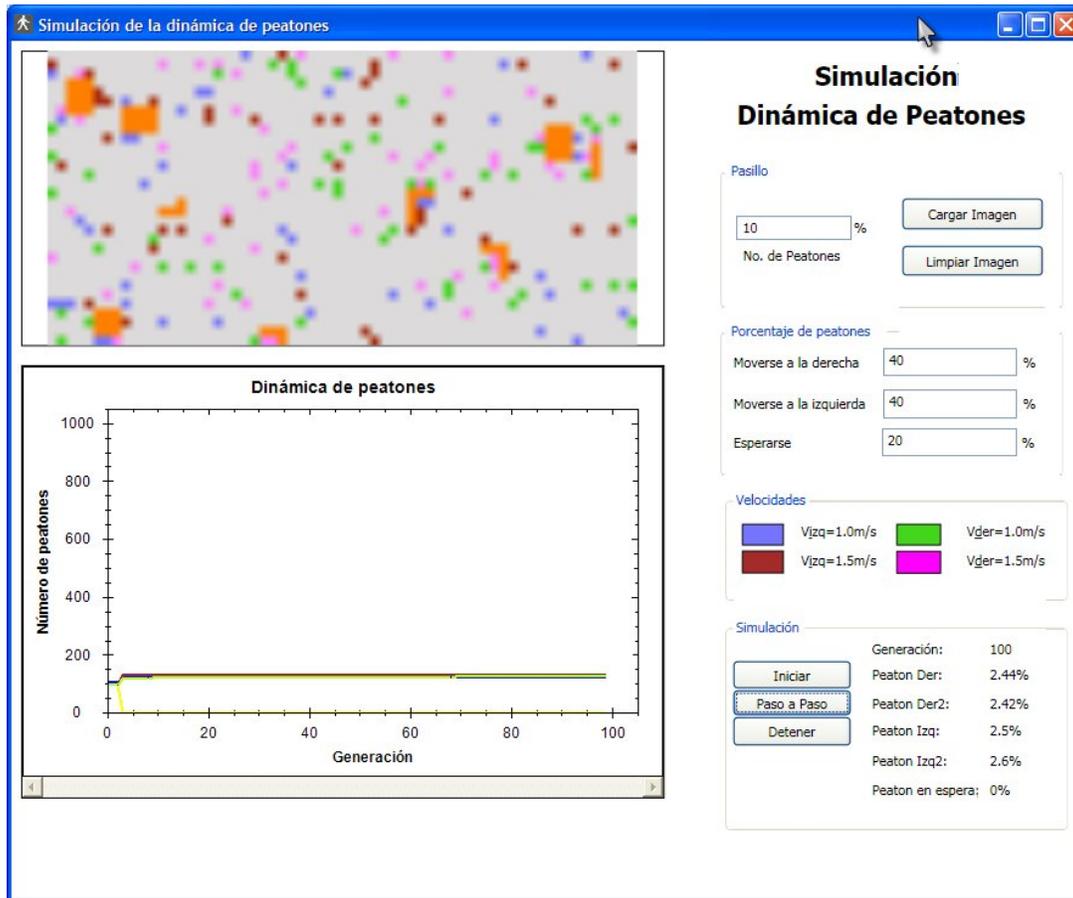


Figura 9. Pantalla que representa el Caso Promedio.

### **CAPÍTULO V. CONCLUSIONES**

Los AC pueden facilitar el estudio de fenómenos físicos, con este proyecto estudiamos el flujo peatonal en un pasillo con diversos obstáculos.

Gracias a este modelo propuesto de Dinámica de Peatones nos damos cuenta que tan complicado es el comportamiento de las personas, dado que no existe una única solución, el sistema se vuelve no determinista, por ello el estudio de este problema mediante el uso de AC.

Este trabajo de Proyecto Terminal no termina aquí, ya que su estudio se puede adaptar a diversas áreas de comportamiento, ya sea supermercados, escuelas, cruces peatonales, etc.

Finalmente los objetivos planteados al inicio de este proyecto se cumplen satisfactoriamente con la simulación diseñada.

## BIBLIOGRAFÍA

- [1] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems* Cambridge University Press, Australia, (1998).
- [2] J. Matsukidaira and K. Nishinari, *Euler-Lagrange correspondence of cellular automaton for traffic-flow models*, Phys. Rev. Vol. 90, (2003).
- [3] H. K. Lee, R. Barlovic, M. Schreckenberg, and D. Kim, *Mechanical restriction versus human overreaction triggering congested traffic states* Phys. Rev. Lett. 92, (2004).
- [4] K. Takimoto and T. Nagatani, *Self-organized phase transitions in cellular automaton models for pedestrian*. Physica A 320, 611 (2003).
- [5] S. Maniccam, *Traffic jamming on hexagonal lattice*. Physica A 321, 653 (2003).
- [6] S. Maniccam, *Congestion of point-to-point mobile objects*. Physica A 331, 669 (2004)
- [7] S. Maniccam, *Effects of back step and update rule on congestion of mobile objects*. Physica A 346, 631 (2005).
- [8] A. Schadschneider, A. Kirchner, and K. Nishinari, *CA Approach to Collective Phenomena in Pedestrian Dynamics*. *Lecture Notes Computer Science*. 2493, 239 (2002).
- [9] W. F. Fang, L. Z. Yang, and W. C. Fan, *Simulation of bi-direction pedestrian movement using a cellular automata model* Physica A 321, 633 (2003).
- [10] A. Kirchner and A. Schadschneider, *Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics* Physica A 312, 260 (2002).
- [11] O. Biham, A. A. Middleton, and D. Levine, *Self organization and a dynamical transition in traffic flow models* Phys. Rev. A 46, R6124 (1992).
- [12] von Neumann, J. (1966)*The Theory of Self-reproducing Automata*, ed. Univ. of Illinois Press, Urbana, IL
- [13] Gardner, M. (1970) *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"*, Scientific American
- [14] Wolfram (1986), *Theory and Application of Cellular Automata*, World Scientific, Singapur

## DOCUMENTOS ANEXOS

Anexo A - Manual de Usuario.

Anexo B - Manual del Programador.

**Universidad Autónoma Metropolitana**

**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

---

**Manual de Programador  
Simulación de la Dinámica de Peatones**



**CBI | INGENIERÍA EN COMPUTACIÓN**

**SIMULACIÓN DE LA DINÁMICA DE PEATONES**

**ALUMNO | FELIPE DE JESÚS RIOFRÍO LEÓN**

**ASESOR | GERMÁN TÉLLEZ CASTILLO**

## TABLA DE CONTENIDO

<b>TABLA DE CONTENIDO</b> .....	<b>- 3 -</b>
<b>REQUERIMIENTOS PARA EL DESARROLLO</b> .....	<b>- 4 -</b>
Hardware .....	- 4 -
Software.....	- 4 -
<b>DIAGRAMAS</b> .....	<b>- 5 -</b>
Diagrama de Casos de Uso.....	- 5 -
Descripción del Diagrama de Casos de Uso.....	- 5 -
Diagrama de Componentes.....	- 6 -
Descripción del Diagrama de Componentes .....	- 6 -
Diagrama de Clases.....	- 7 -
Descripción del Diagrama de Clases .....	- 8 -

## **REQUERIMIENTOS PARA EL DESARROLLO**

Para el desarrollo de este simulador se necesita una computadora con las siguientes características:

### **Hardware**

256 MB en memoria RAM como mínimo  
Procesador Pentium III o superior

### **Software**

SO Windows XP  
Microsoft Visual C# 2008 Express Edition  
Microsoft .NET Framework 3.5  
Microsoft Windows SDK for Visual Studio 2008  
Archivo ZedGraph.dll

*Nota:* el archivo ZedGraph.dll debe de encontrarse dentro de la misma carpeta que contiene al archivo ejecutable "DinamicaPeatones.exe".

## DIAGRAMAS

### Diagrama de Casos de Uso

El siguiente diagrama permite una visión general para lograr entender el funcionamiento de la aplicación.

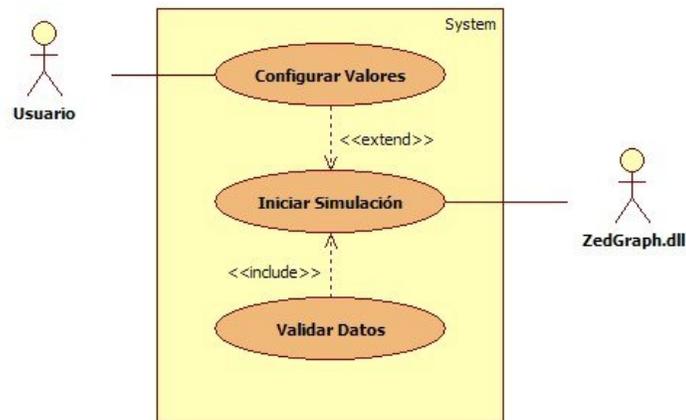


Figura 1. Diagrama de Casos de Uso.

### Descripción del Diagrama de Casos de Uso

- **Configurar Valores:** El caso de uso comienza cuando el usuario coloca sus propios datos en las cajas de texto mostradas en la interfaz. Para iniciar la simulación con esa información el usuario deberá dar *click* en el botón “Iniciar”.
- **Iniciar Simulación:** El caso de uso comienza cuando el usuario presiona el botón “Iniciar”. El sistema carga automáticamente la configuración correspondiente a los valores introducidos anteriormente. Los datos usados para ejecutar la aplicación se muestran en la interfaz. El usuario tiene la posibilidad de personalizar las condiciones iniciales de la simulación, colocando valores a las variables del problema.
- **Validar Datos:** Este caso de uso inicia cuando los datos proporcionados por el usuario son incorrectos, es decir, cuando violan alguna restricción. Se le muestra al usuario el mensaje de error correspondiente.

## Diagrama de Componentes

El siguiente diagrama permite identificar los componentes principales que integran el sistema:

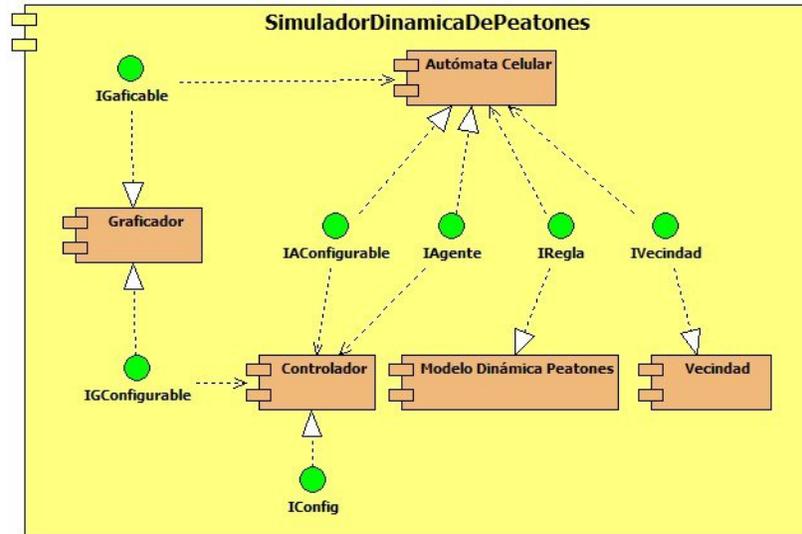


Figura 2. Diagrama de Componentes.

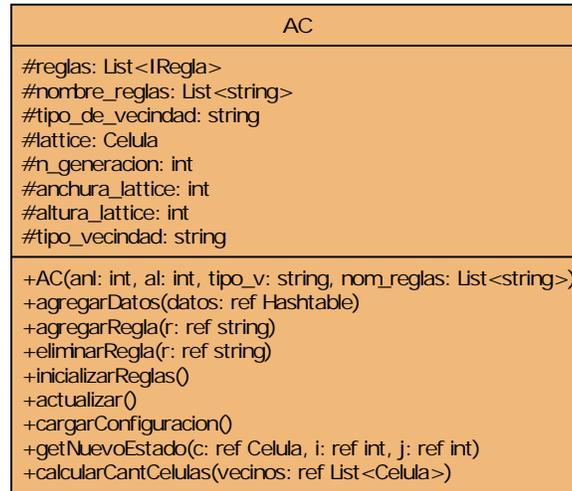
## Descripción del Diagrama de Componentes

- **Autómata Celular:** Contiene las clases que permiten la configuración del autómata celular.
- **Graficador:** Contiene las clases que ejecutan el sistema.
- **Controlador:** Contiene las clases necesarias para administrar la aplicación.
- **Modelo Dinámica Peatones:** Contiene las clases que resuelven el problema de la Dinámica de Peatones.
- **Vecindad:** Contienen las clases que representan la vecindad utilizada para el sistema (vecindad de Moore).

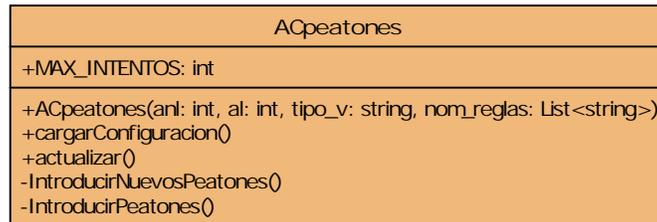


## Descripción del Diagrama de Clases

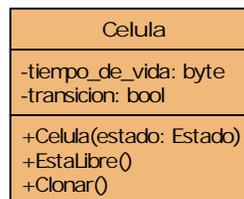
- **AC:** Permite generar las condiciones necesarias para crear el Autómata Celular del sistema.



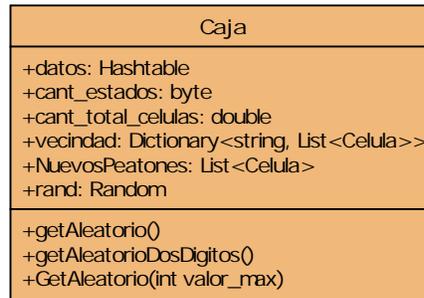
- **ACpeatones:** Configura las condiciones necesarias para crear el Autómata Celular a nivel de Peatón.



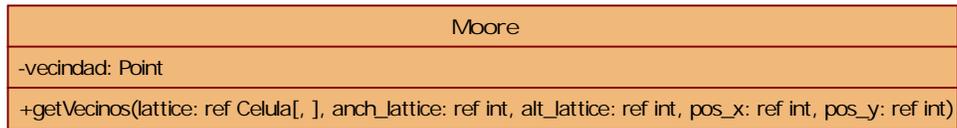
- **Celula:** Contiene las características de cada célula del sistema.



- **Caja:** Genera nuevos peatones de acuerdo dependiendo la situación del sistema.



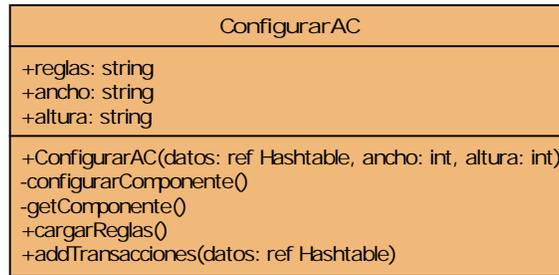
- **Moore:** Genera nuevos peatones de acuerdo dependiendo la situación del sistema.



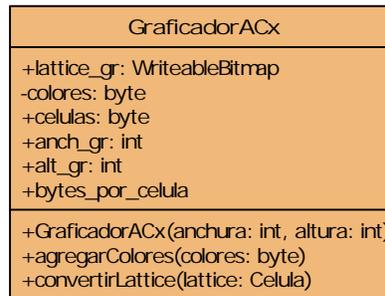
- **Simulador:** Permite llevar a cabo la simulación del sistema así como crear la interfaz gráfica.



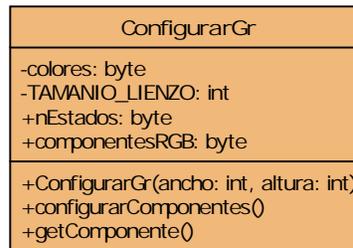
- **ConfigurarAC:** Inicializa el Autómata Celular respecto a las reglas establecidas.



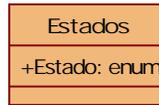
- **GraficadorACx:** Genera los colores de cada peatón de acuerdo a cada iteración .



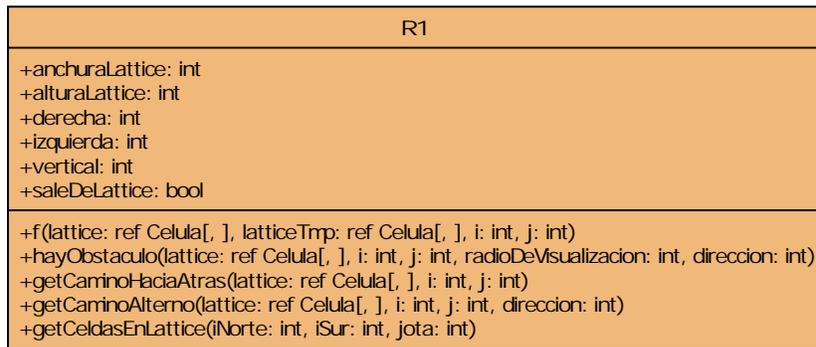
- **ConfigurarGr:** Permite configurar valores iniciales para los colores de cada peatón.



- **Estados:** Identifica el estado en el que se encuentra cada célula del sistema.



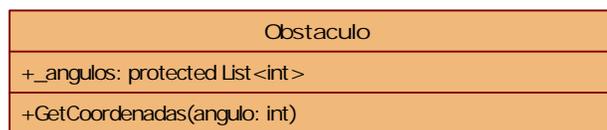
- **R1:** Contiene las reglas de movimiento de acuerdo a cada tipo de peatón.



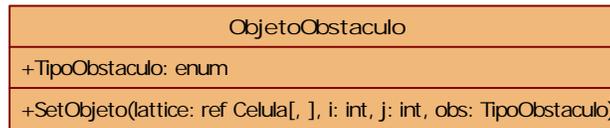
- **R3:** Contiene las reglas de movimiento de acuerdo a cada tipo de peatón.



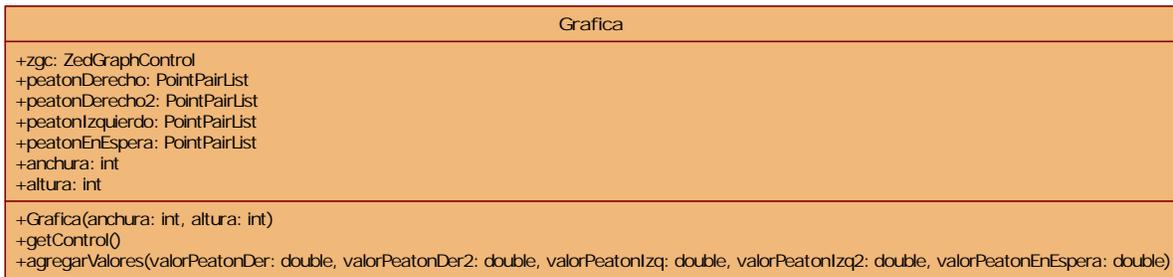
- **Obstaculo:** Contiene las características generales del obstáculo.



- **ObjetoObstaculo:** Genera diversas figuras que representan un obstáculo.



- **Grafica:** Genera la gráfica de la simulación.



**Universidad Autónoma Metropolitana**

**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

---

**Manual de Usuario  
Simulación de la Dinámica de Peatones**



**CBI | INGENIERÍA EN COMPUTACIÓN**

**SIMULACIÓN DE LA DINÁMICA DE PEATONES**

**ALUMNO | FELIPE DE JESÚS RIOFRÍO LEÓN**

**ASESOR | GERMÁN TÉLLEZ CASTILLO**

## TABLA DE CONTENIDO

<b>TABLA DE CONTENIDO .....</b>	<b>- 3 -</b>
<b>REQUERIMIENTOS PARA SU EJECUCIÓN .....</b>	<b>- 4 -</b>
Hardware .....	- 4 -
Software.....	- 4 -
<b>PRIMERA VISTA DEL SIMULADOR .....</b>	<b>- 5 -</b>
Interfaz Gráfica .....	- 5 -
Área de Visualización .....	- 6 -
<i>Pasillo</i> .....	- 6 -
<i>Peatones</i> .....	- 6 -
<i>Obstáculo</i> .....	- 6 -
Área de Grafica.....	- 6 -
Sección Pasillo.....	- 6 -
Sección Porcentaje de peatones .....	- 6 -
Sección Velocidades .....	- 6 -
Sección Simulación.....	- 7 -
<b>EJECUTAR EL SIMULADOR.....</b>	<b>- 7 -</b>
<b>REALIZAR UNA SIMULACIÓN .....</b>	<b>- 7 -</b>
Introducir el número de peatones.....	- 7 -
Configurar variables para el movimiento.....	- 8 -
Cargar imagen del pasillo.....	- 8 -
Iniciar simulación.....	- 9 -
<i>Simulación con timer</i> .....	- 9 -
<i>Simulación paso a paso</i> .....	- 9 -
Observar Sección Velocidades .....	- 10 -
Observar Gráfica.....	- 10 -
Detener simulación .....	- 11 -
Reiniciar simulación.....	- 11 -
<b>EJEMPLO DE UNA SIMULACIÓN CORRIENDO.....</b>	<b>- 12 -</b>

## **REQUERIMIENTOS PARA SU EJECUCIÓN**

Para ejecutar el simulador se necesita una computadora con las siguientes características:

### **Hardware**

Lector de CD  
256 MB en memoria RAM como mínimo  
Procesador Pentium III o superior

### **Software**

SO Windows XP  
Microsoft .NET Framework 3.5  
Archivo ZedGraph.dll

*Nota:* el archivo ZedGraph.dll debe de encontrarse dentro de la misma carpeta que contiene al archivo ejecutable "DinamicaPeatones.exe".

## PRIMERA VISTA DEL SIMULADOR

El simulador es una interfaz gráfica, de tal forma que el usuario pueda utilizarlo de manera intuitiva desde el principio y pueda consultar el comportamiento del flujo de peatones dentro de un pasillo con obstáculos y visualizar la gráfica de peatones existentes de dicho comportamiento.

### Interfaz Gráfica

La interfaz gráfica del sistema (ver *figura 1*), cuenta con seis secciones principales, estas son:

1. Área de Visualización
2. Área de Gráfica
3. Sección Pasillo
4. Sección Porcentaje de peatones
5. Sección Velocidades
6. Sección Simulación

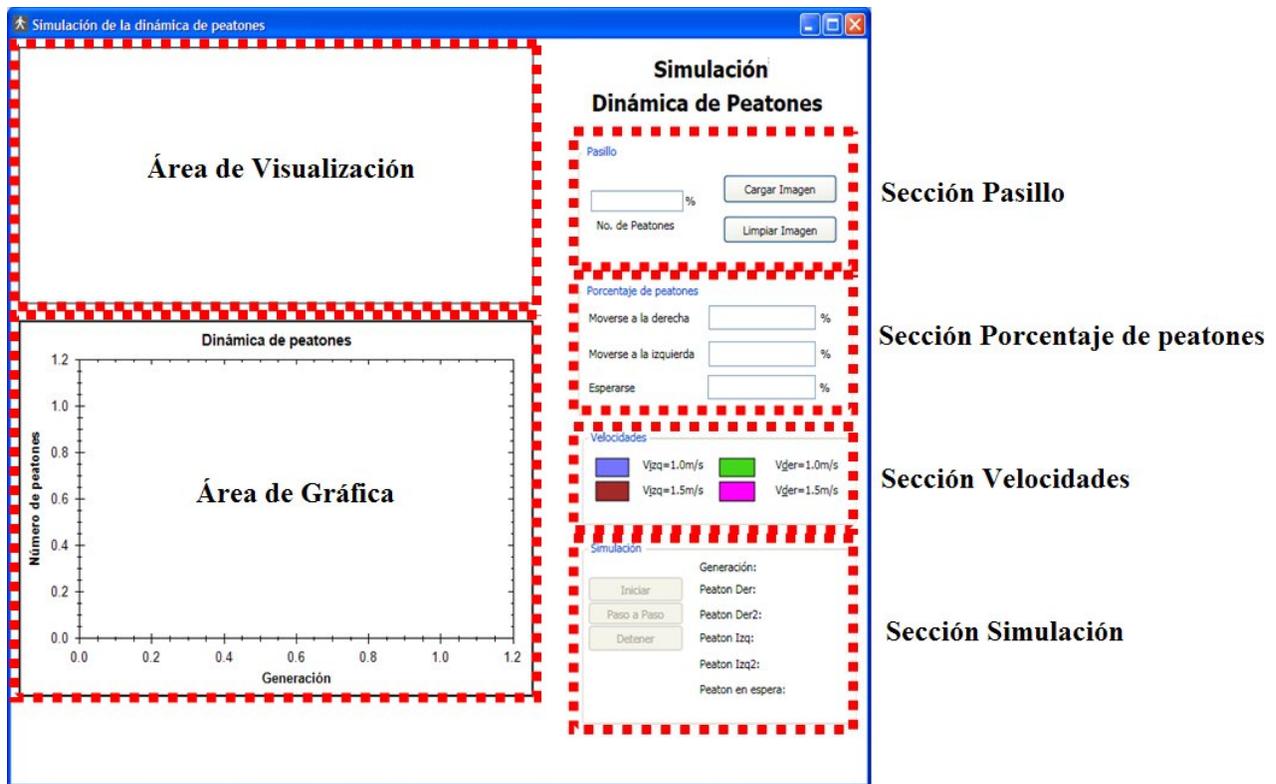


Figura 1. Interfaz Gráfica.

## **Área de Visualización**

Esta área es donde se visualiza la simulación. Se cargan las representaciones gráficas del pasillo así como los cuatro tipos de peatones que interactúan en la simulación.

Dado que cada peatón tiene una velocidad diferente, estos se representan con distintos colores para diferenciar sus diferentes velocidades.

### ***Pasillo***

Es el lugar mediante el cual se realiza la interacción de los peatones durante la simulación. Está representado por un cuadro de color gris y se genera al momento de pulsar el botón “Cargar Imagen”.

### ***Peatones***

Los cuatro tipos de peatones tienen una única velocidad y cada uno de ellos está representado por un *pixel* que está contenido dentro del pasillo.

### ***Obstáculo***

Esta representado por un conjunto de píxeles sin movimiento y que de igual forma está contenido dentro del pasillo (de color anaranjado).

## **Área de Grafica**

En ella se muestra la representación gráfica de lo que sucede en la simulación, es decir, muestra el número de peatones que se encuentran interactuando en el sistema en cada paso del tiempo.

### **Sección Pasillo**

Contiene una cajita de texto que permitirá dar un número (en porcentaje) inicial de peatones para el sistema. Además, contendrá las funciones necesarias para poder crear una configuración inicial (botón Cargar “Imagen”) y a su vez poder eliminar esa misma configuración (botón “Limpiar Imagen”).

### **Sección Porcentaje de peatones**

Contiene las variables que darán origen al número de peatones de los diferentes tipos, esto en forma de porcentajes y tomando en cuenta que el cien por ciento será el número que se eligió previamente en la “Sección Pasillo”.

### **Sección Velocidades**

Contiene una referencia del color acerca de los cuatro tipos de peatones.

## Sección Simulación

Contiene los botones para poder realizar o detener la simulación. El botón “Iniciar” da comienzo a la simulación; el botón “Paso a Paso” permite ejecutar la simulación de forma manual, y el botón “Detener” da una pausa a la simulación pero se podrá reestablecer con el botón “Iniciar”.

## EJECUTAR EL SIMULADOR

Para ejecutar el “Simulador de la dinámica de Peatones” se debe ingresar el CD, ingresar a la carpeta “Simulador” y dar doble *click* en el icono “DinamicaPeatones.exe”, aparecerá la interfaz del simulador.

## REALIZAR UNA SIMULACIÓN

Para crear una simulación de deben de seguir los siguientes pasos:

1. Introducir el número de peatones.
2. Configurar variables para el movimiento.
3. Cargar imagen del pasillo.
4. Iniciar simulación.
5. Observar Sección Velocidades.
6. Observar Gráficas.
7. Detener simulación.
8. Reiniciar simulación.

### Introducir el número de peatones

La cantidad de peatones (en porcentaje) a introducir es en función del total de celdas del pasillo, por ejemplo, si el número introducido es cincuenta (50) esto indica que el pasillo tendrá ocupada la mitad de su área (ver *figura 2*).

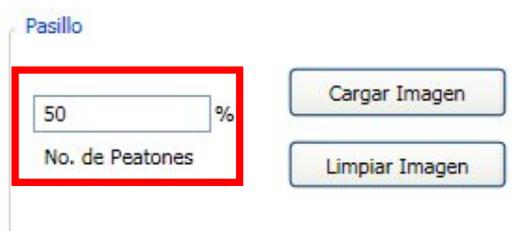
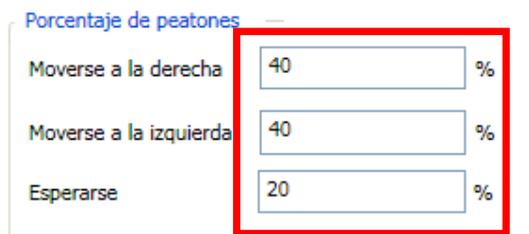


Figura 2. Sección Pasillo.

### Configurar variables para el movimiento

Una vez cargado nuestro entorno grafico donde se generará la simulación, debemos de configurar las variables que rigen la cantidad de los tipos de peatones. Para ello se debe indicar a manera de porcentajes, la cantidad de peatones con movimiento a la derecha y peatones con movimiento a la izquierda, así como la cantidad de peatones que inicialmente tendrán una velocidad de cero y que a la segunda iteración y de forma aleatoria tomarán dichas direcciones (izquierda o derecha) (ver *figura 3*).



Porcentaje de peatones	
Moverse a la derecha	40 %
Moverse a la izquierda	40 %
Esperarse	20 %

Figura 3. Sección Porcentaje de Peatones.

### Cargar imagen del pasillo

Después de haber indicado el número de peatones y de haber configurado las variables, debemos cargar la imagen del pasillo. Para cargarla dentro del área de visualización debemos de dar un *click* en el botón llamado “Cargar Imagen” (ver *figura 4*).



Pasillo	
50 %	Cargar Imagen
No. de Peatones	Limpiar Imagen

Figura 4. Sección Pasillo (cargar imagen).

El pasillo quedará inicializado junto con los diversos tipos de peatones configurados previamente (ver *figura 5*).

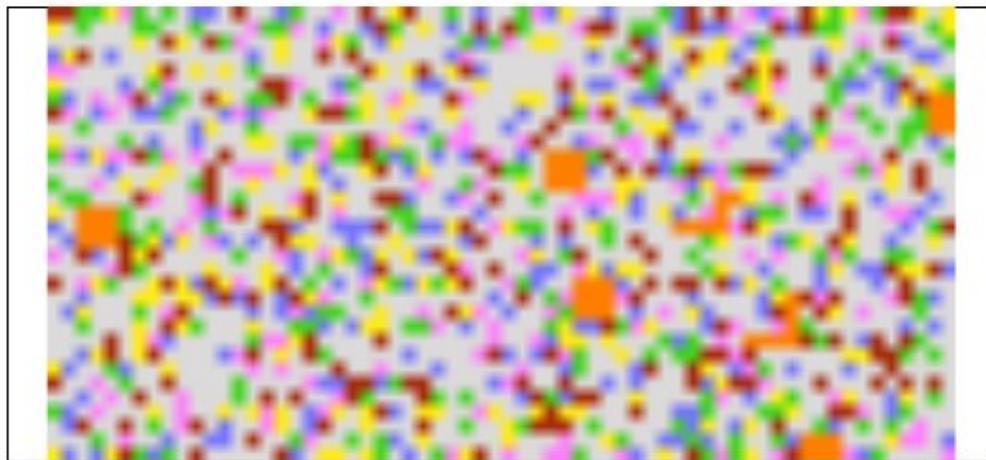


Figura 5. Área de Gráfica.

## Iniciar simulación

Una vez configurados todos los parámetros podemos dar comienzo a la simulación, esto puede realizarse de dos formas diferentes, una realizar la simulación automáticamente o realizarla paso a paso.

### *Simulación con timer*

La simulación con *timer* es una simulación donde cada generación de la simulación es controlada por un *timer* dentro del sistema; es una simulación cuyos pasos o evoluciones son automáticos. Para realizar este procedimiento basta con dar un *click* en el botón “Iniciar” y comenzará a verse dentro del área de visualización el comportamiento de los peatones dentro del pasillo (ver *figura 6*).

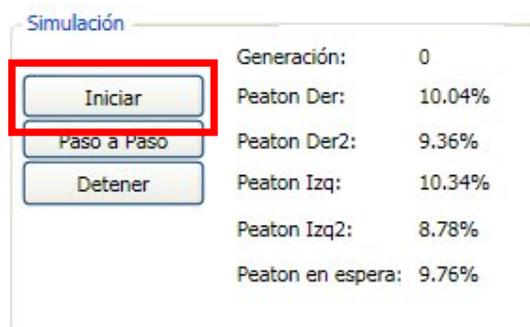


Figura 6. Sección Simulación (timer).

### *Simulación paso a paso*

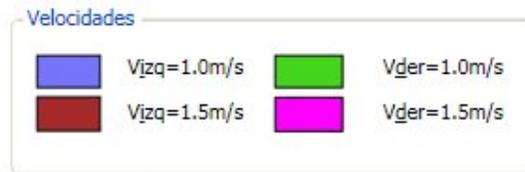
La simulación paso a paso es una simulación donde cada generación es controlada por el usuario. Es una simulación cuyos pasos o evoluciones son realizados en forma manual. Para realizar este procedimiento basta con dar un *click* en el botón “Paso a Paso” y se observará en el área de visualización una sola generación del movimiento de los peatones dentro del pasillo (ver *figura 7*).



Figura 7. Sección Simulación (paso a paso).

### Observar Sección Velocidades

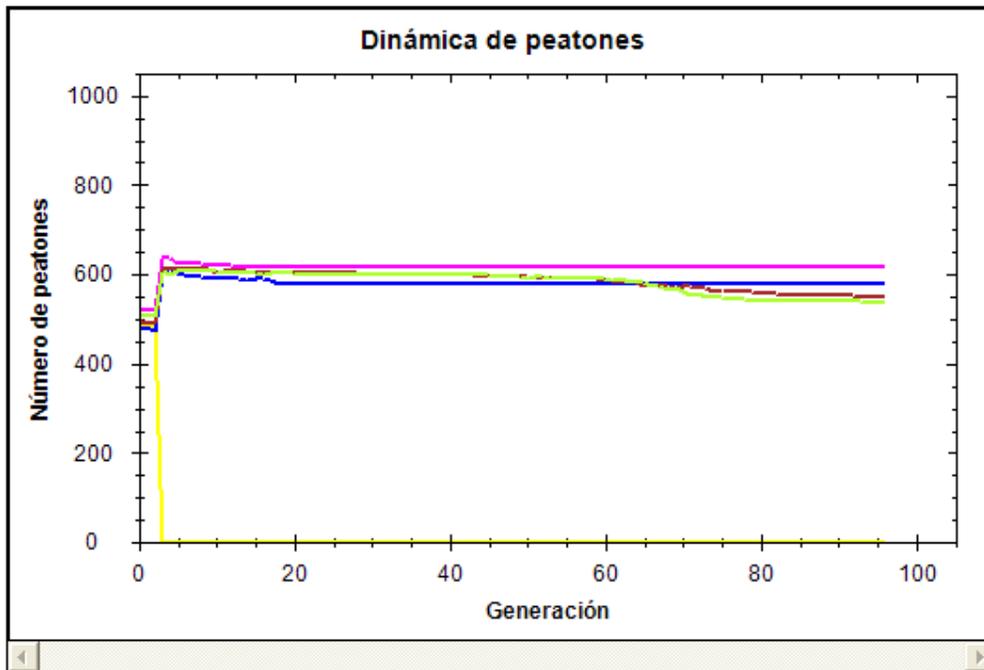
Se podrá identificar cada tipo de peatón por el color asignado (ver *figura 8*).



**Figura 8.** Sección Velocidades. Tipos de peatones y su respectiva velocidad..

### Observar Gráfica

Este simulador realiza una gráfica que muestra la cantidad total de los diferentes tipos de peatones (ver *figura 9*).



**Figura 9.** Área de Gráfica. Los colores indican el tipo de peatón.

### Detener simulación

Para detener una simulación solo basta con dar un *click* en el botón “Detener” (ver *figura 10*).

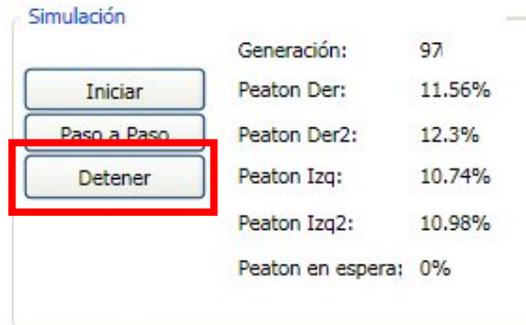


Figura 10. Sección Simulación (detener).

### Reiniciar simulación

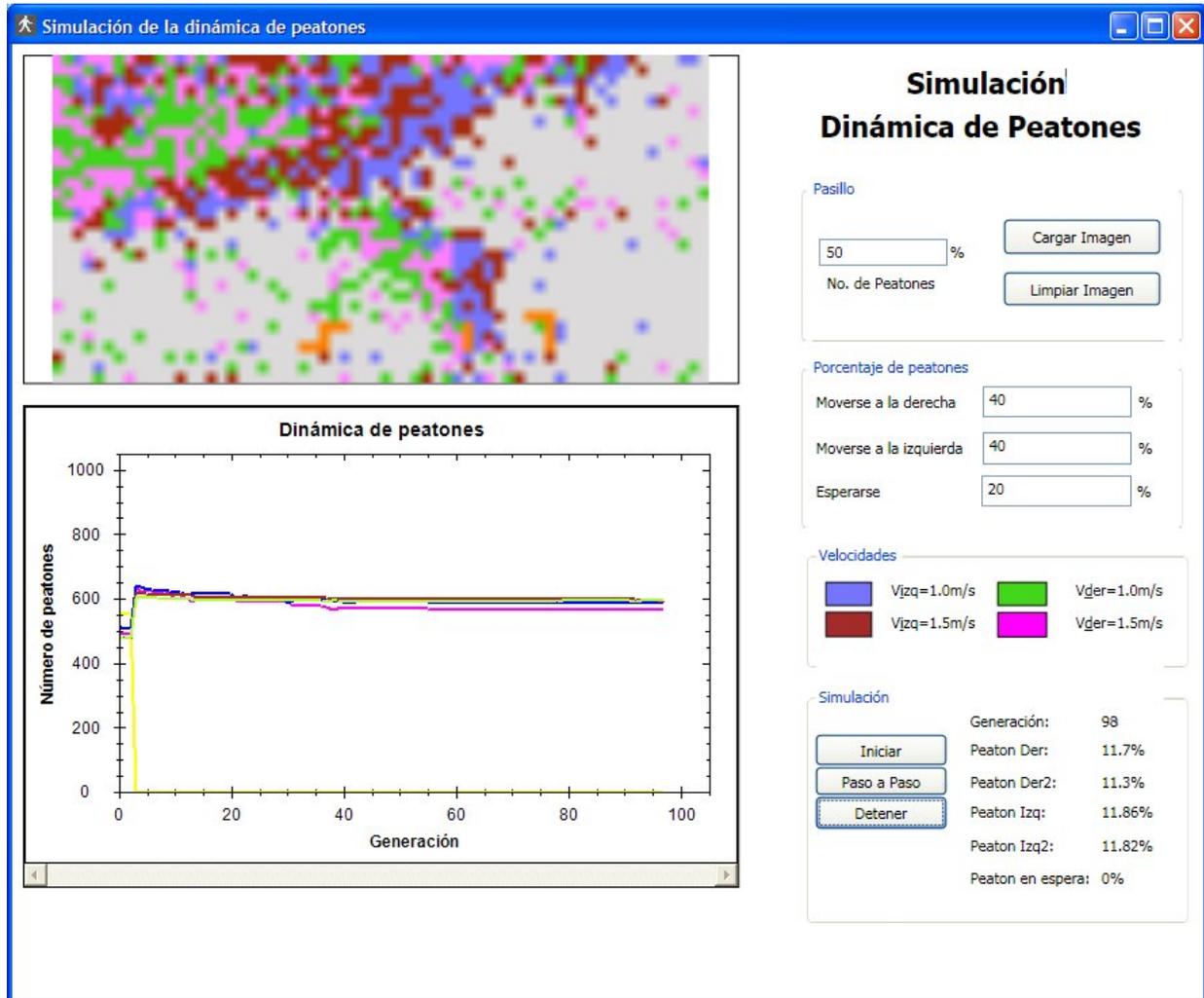
Para reiniciar una simulación desde el tiempo cero sin peatones, se debe de dar un *click* en el botón “Limpiar, e iniciar la Simulación nuevamente (ver *figura 11*).



Figura 11. Sección Pasillo (limpiar imagen).

## EJEMPLO DE UNA SIMULACIÓN CORRIENDO

Esta simulación ejemplo contiene los parámetros que se observan en las imágenes de este documento, en ella podemos ver el comportamiento de la interacción de los peatones dentro de esta simulación junto con la gráfica que genera (ver *figura 12*).



**Figura 12.** Simulación del sistema generado en este proyecto terminal.