

1er Proyecto de Simulación:

## Poblado en Evolución

Miguel Tenorio Potrony



Facultad de Matemática y Computación  
Universidad de la Habana

Curso 2019/2020

# Índice

<b>1. Generales del Estudiante</b>	<b>3</b>
<b>2. Orden del Problema Asignado</b>	<b>3</b>
<b>3. Principales ideas seguidas para la solución del problema</b>	<b>3</b>
3.1. Flujo del tiempo . . . . .	3
3.2. Implementación de los eventos . . . . .	3
<b>4. Modelo de Simulación de Eventos Discretos desarrollado para resolver el problema</b>	<b>5</b>
<b>5. Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema</b>	<b>8</b>
<b>6. Enlace al repositorio del proyecto en Github</b>	<b>9</b>

## 1. Generales del Estudiante

- **Nombre y apellidos:** Miguel Tenorio Potrony
- **Grupo:** 412
- **Email:** [m.tenorio@estudiantes.matcom.uh.cu](mailto:m.tenorio@estudiantes.matcom.uh.cu)

## 2. Orden del Problema Asignado

- **Poblado en Evolución**

## 3. Principales ideas seguidas para la solución del problema

En la realización de la simulación se pueden definir dos cuestiones importantes de cierta dificultad: el flujo del tiempo, y la implementación de los eventos.

### 3.1. Flujo del tiempo

Para su resolución, se generan inicialmente los períodos o turnos en forma de semanas, de modo que el primero de estos es la semana 0 y, el que define la ocurrencia del año 1, es la semana 48. De este modo, una iteración del proceso da paso a un turno.

En la generación de estos períodos se tomó en cuenta usar una variable aleatoria uniforme (o un proceso de Poisson homogéneo) dado que no hay especificaciones que, sugieran el uso de una función de intensidad para un proceso de Poisson no homogéneo. En la ejecución del programa se puede escoger con cuál de estas formas se quiere simular el flujo del tiempo, incluso se pueden definir sus parámetros.

### 3.2. Implementación de los eventos

En orden de exponer las ideas tomadas, se hace necesario definir los **eventos**:

1. **Fallecimiento**
2. **Nacimiento**
3. **Tiempo de soledad**
4. **Deseo de tener pareja**

## 5. Establecimiento de pareja

## 6. Ruptura de pareja

## 7. Embarazo

De los eventos antes mencionados, los primeros tres son generados con una fecha de ocurrencia.

A continuación se explica cómo se implementaron dichos eventos, exponiendo la forma en la que se crean y las consecuencias que se derivan de su ocurrencia.

**Fallecimiento** Las personas tienen una característica llamada *inmunidad*, la cual les permite estar un tiempo libres de este evento. Si alguien no tiene inmunidad, se genera la probabilidad de morir según su edad, siguiendo las probabilidades de la orientación del proyecto. Este proceso tiene como resultados la posibilidad de que la persona pueda o no morir, y un tiempo  $t$ . En caso de que esta vaya a morir, se agrega a la cola de eventos un evento "fallecimiento" para esta persona con fecha  $tiempo\_actual + t$ . En caso contrario, se le agrega  $t$  a la inmunidad de ese sujeto.

Si se procesa un evento "fallecimiento" sobre una persona  $\chi$ , se comprueba que  $\chi$  tenga pareja, de modo que, de tenerla, dicha pareja envíe y se genere un evento *Tiempo de soledad*. En términos de variables de estado (SS en inglés; están detalladas en la sección de modelo de simulación desarrollado), se aumenta en 1 el contador de muertes y se desmarca  $\chi$  como persona viva.

**Deseo de tener pareja** Este evento actualiza las SS de forma inmediata, ya que no necesita una fecha de ocurrencia. A todas las personas que tienen más de 12 años y no tienen pareja, se les calcula la probabilidad de que deseen tener pareja según su edad; en caso que "deseen" tenerla, se le agrega a una cola de solteros que desean pareja, dado su sexo.

**Establecer pareja** Cabe aclarar que, por razones -no ideológicas- de simplicidad e implementación y , se forman parejas solo entre hombres y mujeres. Se tomaron en cuenta dos vías para implementar este suceso:

1. Dados dos grupos de personas que deseen tener pareja, uno de mujeres y otro de hombres, tratar de emparejar a aquellos miembros de diferentes grupos tal que su diferencia de edad máxima sea de 5 años. Los que queden sin pareja se tratan de emparejar con los que tengan de 5 a 10 años de diferencia de edad. Repetir este proceso según los grupos de diferencia de edad dados en la orientación. Al final del proceso, los que queden sin pareja, se quedan solteros para la próxima iteración. Este método trata de acercarse a la realidad, donde generalmente las

personas forman pareja con aquellas con las cuales su diferencia de edad no es mucha.

2. Mismos grupos que en la Vía 1, pero primero se les aplica un orden aleatorio. A los miembros de un grupo se les trata de emparejar uno a uno con los del otro. Los que queden sin pareja, se quedan solteros para la próxima iteración.

Es evidente que esta vía es más sencilla que su antecesora, pero en la práctica trajo buenos resultados; por tanto, fue la escogida.

**Ruptura de pareja** Por cada pareja se genera la probabilidad de que se separen, por la probabilidad dada en la orientación. En caso de que ocurra la ruptura, se generan los eventos *Tiempo de soledad* para cada una de las personas.

**Tiempo de soledad** Se puede apreciar, dados los anteriores eventos, que este suceso se genera cuando en una pareja uno de sus miembros fallece o ambos deciden separarse. Cuando algo de esto ocurre, se genera un tiempo  $t$  -para cada persona de la pareja o para la que quedó viuda- con el cual se puede determinar la fecha hasta la que cada afectado, no desearía pareja.

Si se procesa un evento "Tiempo de soledad" sobre una persona  $\chi$ , se comprueba que  $\chi$  esté viva; de estarlo,  $\chi$  termina su tiempo de soltería y vuelve a ser sensible al evento *Deseo de tener pareja*.

**Embarazo** El proceso a través del cual una mujer queda embarazada se sigue al pie de la letra con respecto a la orientación y con el detalle de que se intenta definir, desde el inicio de la gestación del embarazo, si este será múltiple o no y, en caso de serlo, cuántos bebés nacerían. Acorde a las restricciones, si todo está en orden, se genera la probabilidad de que la mujer quede embarazada y, de lograrlo, se genera un evento *nacimiento* con fecha  $tiempo\_actual + 36$ , ya que 36 representa 9 meses en semanas.

**Nacimiento** Al ocurrir este suceso, se comprueba que la madre esté viva y, de estarlo, se determina el sexo del bebé o de los bebés.

## 4. Modelo de Simulación de Eventos Discretos desarrollado para resolver el problema

El modelo implementado se encuentra inspirado en el modelo de un sistema de atención de un solo servidor. Consiste en un generador de eventos donde los sucesos que tienen fecha  $f$  (tal como se definieron anteriormente), son introducidos a una cola de eventos ordenados por su momento de ocurrencia (figura 1). De este modo, siempre que  $f \in [fecha\_actual, siguiente\_turno]$ , entran al servidor dichos eventos.

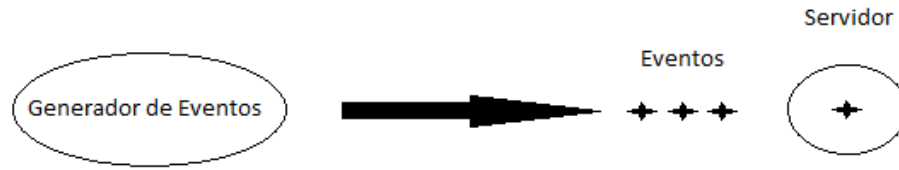


Figura 1: Modelo de Simulación

Pasemos a definir las variables usadas (se usará la terminología usada en la implementación para un mejor entendimiento del programa).

### 1. Variables del tiempo

- *actual\_time*: Representa la fecha de la iteración actual
- *turns*: Lista que mantiene la secuencia de períodos restantes sobre los cuales se realizarían las iteraciones del proceso de simulación.

### 2. Variables contadoras

La simulación es capaz de ejecutarse de forma tal (véase README del proyecto) que se puedan apreciar para cada iteración del proceso los siguientes datos:

- Cantidad de personas vivas por sexo y rango de edad,
- Número de partos pendientes,
- Total de personas vivas,
- Número de muertes y nacimientos ocurridos en la iteración.

El programa se puede acomodar fácilmente para mantener esta información en una o varias variables, según la comodidad del interesado. No obstante, no se estimará necesario, ya que toda esta información puede escribirse en un archivo si el usuario así lo desea. Toda la explicación se debe a que se trata de ser fiel la formalidad mostrada en las clases y en la bibliografía [1].

### 3. Variables de estado del sistema

- *people*: Lista de las personas que conforman la población. Cada persona mantiene los siguientes datos:

- a) *sex*: Sexo de la persona,
- b) *age*: Edad de la persona en semanas,

- c) *childrenNumber*: Cantidad de hijos que desea tener (si esta persona llega a tener un hijo en algún momento, por ejemplo, esta cantidad es restada en uno),
  - d) *immunity*: Inmunidad referida en la descripción del evento *fallecimiento*,
  - e) *married*: Describe si la persona tiene pareja o no,
  - f) *alive*: Define si la persona está viva o no
- *couples*: Lista de las parejas actuales,
  - *newSingles*: Lista de las personas que no tienen pareja, y no desean todavía tenerla, sin encontrarse cumpliendo un tiempo de soledad,
  - *singles\_i*: Lista de personas de sexo  $i$  que no tienen pareja, pero desean tenerla.  $i \in M, F$ .
4. *events*: Representa la cola de eventos con fecha. Dada la naturaleza de la generación de los eventos con fecha, los cuales se generan usando uniformes o exponenciales en dependencia del caso. Esta cola se necesita ordenar cada vez que haya que procesar, o sea, el problema no es exactamente igual a los modelos de simulación de eventos discretos vistos en clases, donde se cumple la invariante dada la cola de eventos  $EL$  y dos sucesos con tiempos  $t_i$  y  $t_j$ , donde  $i$  y  $j$  representan la posición de ellos en  $EL$ ,  $t_i > t_j$  si  $i > j$ .

Describamos ahora una iteración  $i$  del proceso:

1. Se generan los posibles fallecimientos con sus fechas.
2. Se generan las rupturas posibles con sus fechas.
3. Se procesan los eventos pertenecientes a *events* correspondientes al turno actual.
4. Se generan los deseos de tener pareja.
5. Se forman las parejas que se puedan formar.
6. Se generan los embarazos posibles.
7. Se actualiza la fecha actual, las edades de las personas vivas, sus inmunidades (en caso de poseerlas). Es aquí donde se comprueba si se ha cumplido el proceso completo de simulación (100 años segun la orientación).

## 5. Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema

Es válido aclarar que, gracias a la inmunidad que se le otorga a las personas cuando sobreviven a un evento *fallecimiento*, es posible hacer una simulación de las vidas de las personas con una tasa de mortalidad adecuada para un modelo que pretende simular una sociedad moderna y sencilla sin la presencia de sucesos especiales como catástrofes naturales, pandemias, guerras, ataques alienígenas, etc.

El flujo estable de la población, viene dado por una tasa de natalidad, la cual se puede ver afectada por el modo en el que se realiza la simulación. Esto quiere decir que, si el promedio de la diferencia entre turnos es muy grande, en el orden de varios años, la mortalidad no se ven tan afectada por la natalidad, ya que, asumiendo que una pareja desee tener varios hijos, se le estaría restringiendo a tener hijos solo cada ciertos años, cuando podría tenerlos cada año. Incluso un miembro de la pareja podría morir, lo cual implica que dado la frecuencia larga de los turnos se demore más el sobreviviente en encontrar una nueva pareja.

Por tanto si se quiere encontrar un comportamiento más realista, se recomienda simular el tiempo en períodos con un promedio máximo de un año. Para promedios inferiores es capaz de hacer la simulación más cercana a la realidad, con la evolución normal de la mayoría de las poblaciones en la historia, con tendencia al aumento de la población o a su estabilidad, no a su extinción ni a un envejecimiento absoluto y generalizado durante décadas. Dicha evolución contempla años donde hubo más fallecimientos que nacimientos y viceversa.

Los factores que determinan la rapidez del proceso son la cardinalidad de la población y la frecuencia promedio de los turnos. Mientras mayor sea la población, más procesamiento es requerido, al igual que si la frecuencia promedio de los períodos es pequeña, ya que esto se reduce en un mayor número de iteraciones.

Con una población inicial de menos de 10 000 personas y turnos de 6 meses a un año como promedio, se pueden obtener resultados realistas y rápidos. Se hicieron pruebas con diferentes proporciones de población respecto al sexo. Cuando la población inicial de mujeres es igual o cercana a la de los hombres, se percibe un crecimiento de casi el doble de la población inicial para el fin de la simulación. Si la proporción de hombres es mucho mayor que de mujeres (20/80), se observa un flujo poblacional estable, con un ligero crecimiento. Cuando esto ocurre al revés, se nota lo mismo, solo que con menor crecimiento. En ambos casos, los dos grupos poblacionales tienden a equilibrarse. Con mayores desproporciones se observa un decrecimiento de la



población. Puede resultar interesante que cuando la mayor proporción es de hombres, el decrecimiento es menor.

En el README del proyecto se sugieren algunos parámetros para simular el tiempo usando un proceso de Poisson homogéneo, con el valor esperado del tiempo para cada uno. Si se quiere una mayor dispersión de los turnos dada una frecuencia promedio  $f$ , se recomienda usar el proceso de Poisson en vez de una variable uniforme.

## 6. Enlace al repositorio del proyecto en Github

<https://github.com/stdevAntiD2ta/Town-Evolution/tree/master>

## Referencias

- [1] Sheldon M. Ross. Simulation, 5th Edition, capítulo 7, epígrafe 7.1, página 111.