

Simulación basada en sistemas multiagentes - Juego de la vida

Ingeniero Helver Valero Augusto Bustos

Didier Alejandro Bernal Velandia

Estudiante

Universidad Pedagógica Y Tecnológica de Colombia

Facultad de ingeniería

Ingeniería de sistemas y computación

simulación por computadores

primer semestre

2022

INTRODUCCIÓN

En la siguiente documentación se presentará una aproximación a lo que se le denominará el juego de la vida, se habla de una aproximación porque comparado al juego de la vida que propuso el matemático John Horton Conway quien desarrolló un algoritmo donde se aplica la simulación basada en sistemas multiagentes, entonces, la simulación propuesta consiste en parametrizar ciertas características que un cuerpo puede poseer como lo son:

- edad
- Altura
- género
- posición en x
- posición en Y
- año de nacimiento

Otra característica que se le cambia comparado con el modelo de Conway es que ahora hay un espacio donde los cuerpos son liberados y no se ubican en una matriz. La simulación que se propone mostrará resultados sustentando los comportamientos que tiene determinada población en un espacio controlado y un par de años asignados.

DESARROLLO

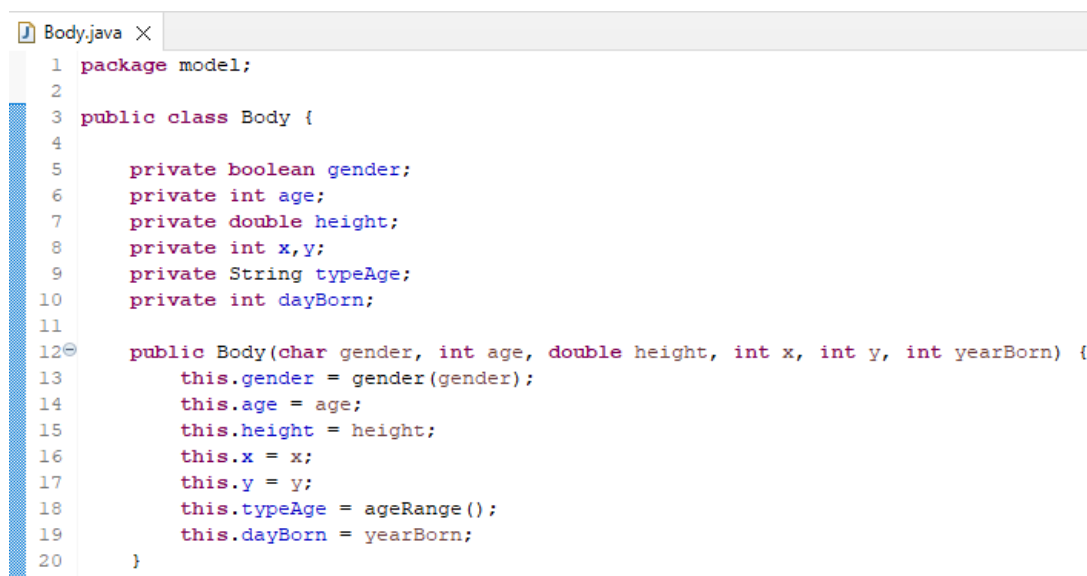
Para tener en cuenta y en consideración: Esta simulación está basada en sistemas multiagentes y se desarrolla bajo el lenguaje de programación java orientado a objetos, para la codificación se hace el uso del entorno de desarrollo gratuito ECLIPSE .



IMAGEN N°1 Entorno de desarrollo Eclipse

Como primer paso de la propuesta de simulación se tiene que un cuerpo tiene diversos parámetros, los cuales en este caso se hacen es importante.

- Creamos una clase Body: a esta clase se le asignan los siguientes atributos y se crea un constructor donde parametrizamos la clase.
 - Gender
 - Altura
 - género
 - posición en x
 - posición en Y
 - año de nacimiento



```
1 package model;
2
3 public class Body {
4
5     private boolean gender;
6     private int age;
7     private double height;
8     private int x,y;
9     private String typeAge;
10    private int dayBorn;
11
12    public Body(char gender, int age, double height, int x, int y, int yearBorn) {
13        this.gender = gender(gender);
14        this.age = age;
15        this.height = height;
16        this.x = x;
17        this.y = y;
18        this.typeAge = ageRange();
19        this.dayBorn = yearBorn;
20    }
21 }
```

IMAGEN N°2 Clase Body parametrizada

continuyendo con la codificación se realiza un método donde se le sumará un año al cuerpo que nació en medio de la simulación, esto con el fin de que el agente crezca y haga cumplir las condiciones que se le asignan por medio del año que se le asigno por parametro, en la siguiente imagen se representa la codificación realizada:

```
public void aumentandoEdad(int year) {
    if (year > 0) {
        age++;
    }
}
```

IMAGEN N°3 Método aumentandoEdad()

ahora, con el siguiente método representado en la imagen se le asignara el sexo o tipo de cuerpo al agente, este tipo se le asignara por medio de un char que sera parametrizado, queda de la siguiente manera:

```
public boolean gender(char string) {  
    if(string == 'M') {  
        gender = false;  
    } else if (string == 'F') {  
        gender = true;  
    }  
    return gender;  
}
```

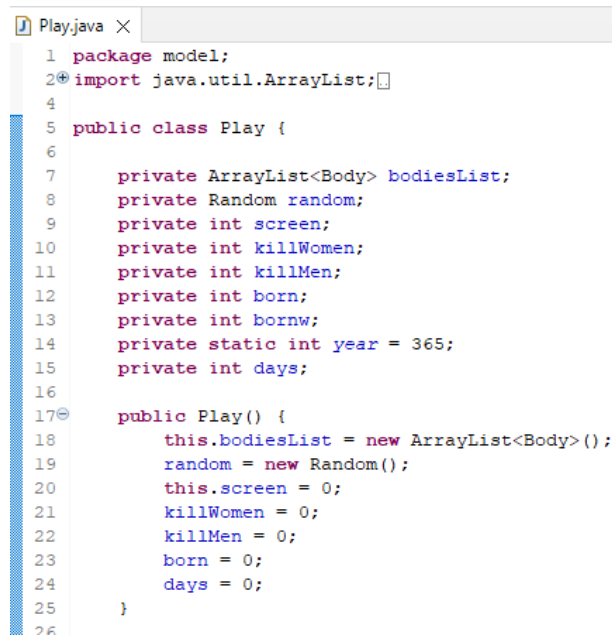
IMAGEN N°4 Método gender()

Con el siguiente método se clasificaron las etapas de los agentes dependiendo de la edad, se identificara si es un niño, joven, adolescente, adulto o anciano.

```
public String ageRange() {  
    String type = "";  
    if (age >= 0 && age <= 11) {  
        type = "Niñez";  
    } if(age >= 12 && age <= 18) {  
        type = "Adolecente";  
    } else if(age >= 14 && age <= 26) {  
        type = "Joven";  
    } else if (age >= 27 && age <= 59) {  
        type = "Adulto";  
    } else if(age >= 60) {  
        type = "Anciano";  
    }  
    return type;  
}
```

IMAGEN N°5 Método ageRange()

- Ahora para iniciar con la simulación creamos la clase Play() donde se codifica los respectivos métodos para que la simulación sea un éxito, la siguiente imagen ilustra la clase mencionada anteriormente:



```

1 package model;
2 import java.util.ArrayList;
3
4
5 public class Play {
6
7     private ArrayList<Body> bodiesList;
8     private Random random;
9     private int screen;
10    private int killWomen;
11    private int killMen;
12    private int born;
13    private int bornw;
14    private static int year = 365;
15    private int days;
16
17    public Play() {
18        this.bodiesList = new ArrayList<Body>();
19        random = new Random();
20        this.screen = 0;
21        killWomen = 0;
22        killMen = 0;
23        born = 0;
24        days = 0;
25    }
26

```

IMAGEN N°6 Clase Play()

En esta clase se harán los cálculos donde

- Se aumentará la edad de los cuerpos respecto a los años que se parametrizan.
- añadimos cuerpos al juego
- se crean los cuerpos dependiendo el tipo o género del agente
- Se clasifican los tipos de etapas de los agentes.
- Método donde se condiciona el comportamiento de dos cuerpos cuando comparten el mismo espacio.
- método donde los cuerpos destruyen a otro.
- Método de la generación de un cuerpo, nacimientos.
- cantidad de hombres.
- cantidad de mujeres
- calcular los días por la cantidad de años que se ingresaron
- darle tamaño al espacio donde los agentes inician su vida reproductiva o supervivencia.

todos estos cálculos y métodos mencionados están presentes en el archivo del juegoDeLaVidaSP.zip

ANÁLISIS Y RESULTADOS

Con los datos parametrizados obtenemos información de la simulación realizada con los cuales se generan una par de gráficas que ilustran el comportamiento de los agentes con respecto al tiempo que se genera, continuando con el desarrollo la simulación queda de la siguiente manera:

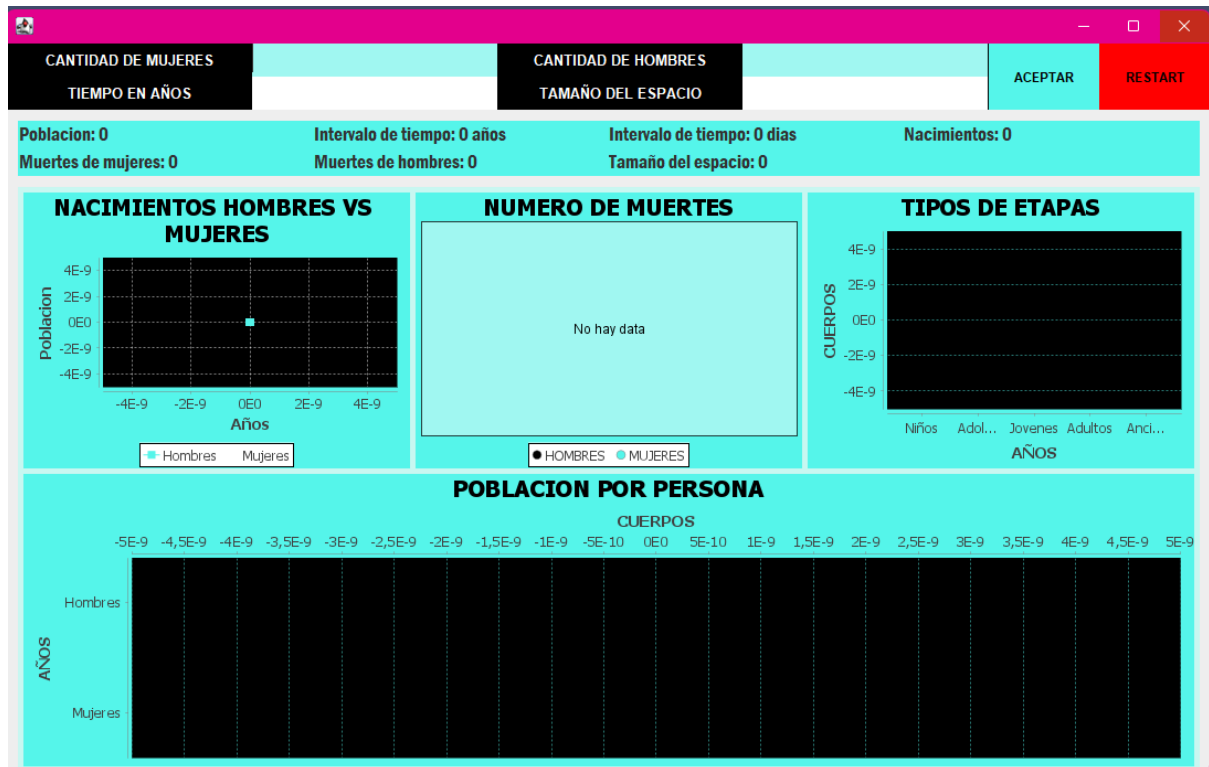


Imagen N°7 Interfaz principal de la simulación juego de la vida

En la interfaz podemos observar que tenemos unos apartados de gráficas donde los resultados arrojados por los parámetros que se ingresaron de la simulación nos darán información respecto a:

- La cantidad de agentes que nacieron o que fueron creados a partir de la condición en la que dos cuerpos o agentes de diferente género o tipo se encuentran en un mismo punto del espacio. Gráfica 1 de nacimientos



Imagen N°8 Interfaz gráfica de Nacimientos Hombres vs Mujeres

- En la que dos agentes del mismo género o tipo se encuentran por medio de la condición en la que sí comparten el mismo punto el más joven morirá por cuestiones de pleito por el espacio. Esta se almacena en la gráfica de número de muertes.

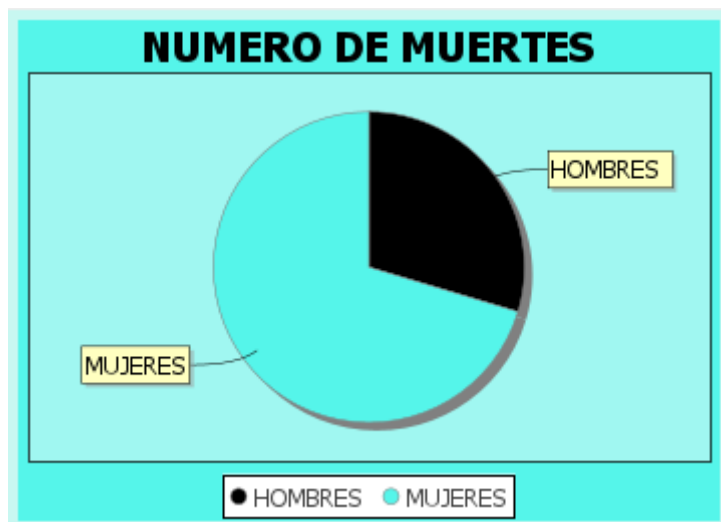


Imagen N°9 Interfaz gráfica del número de muertes

- Se clasifican los agentes por sus tipos de etapas de crecimiento, la gráfica de tipos de etapa.

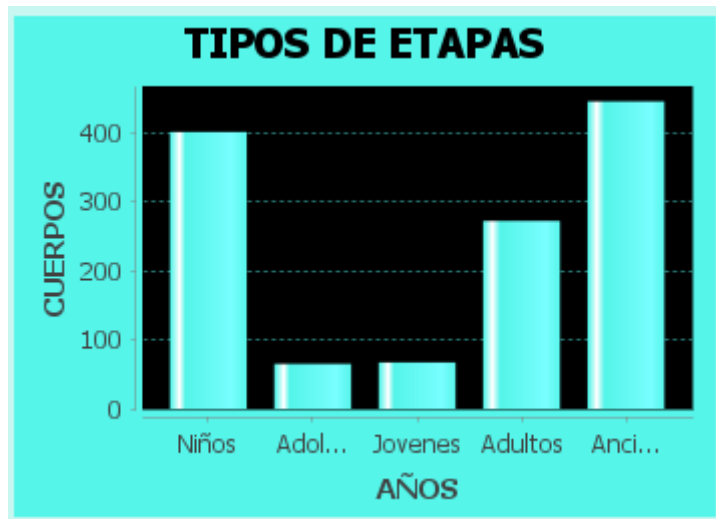


Imagen N°10 Interfaz gráfica de Tipos de etapa de los agentes

- por último se presenta la gráfica de la población por persona, donde se clasifican la cantidad de hombre y mujeres, incluyendo los hombres y mujeres que nacieron.



Imagen N°11 Interfaz gráfica de la cantidad de hombres vs mujeres

1. Para pruebas de simulación, se ingresarán
 - 556 = Mujeres
 - 426 = Hombres
 - 20 = Años
 - Tamaño del espacio = 100
2. Tenemos que la cantidad de nacimientos fueron de 332 y se puede representar en la siguiente gráfica



Imagen N°12 Interfaz gráfica de la cantidad nacimientos por genero

3. Con el siguiente apartado nos dirigimos a la cantidad de muertes que se presentaron en la población, en la gráfica se ilustra lo mencionado anteriormente:



Imagen N°13 Interfaz gráfica de la cantidad de muertes por genero

4. ahora tenemos la cantidad de personas que se clasifican por etapas, con la información recogida por los parámetros mencionados anteriormente obtenemos que

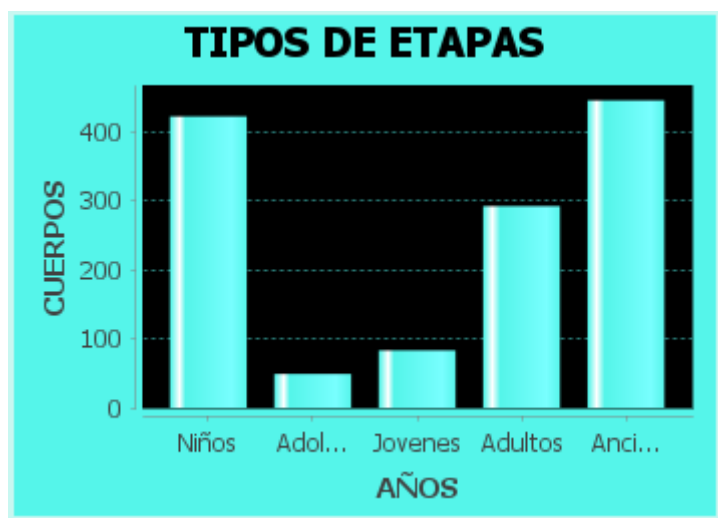


Imagen N°14 Interfaz gráfica de la cantidad de la clasificación de las etapas

5. Pasamos a la gráfica de barras horizontales donde nos muestra como es el comportamiento actual de la población de los hombres y las mujeres, aquí también se incluyen los nuevos agentes que surgieron o nacieron, a lo que se podrá notar que el incremento de las mujeres es mucho mayor que la de los hombres.



Imagen N°15 Interfaz gráfica de la población actual de los dos generos

MANUAL DE USO

Para ejecutar el programa es necesario contar con el JDK instalado de java.

- **NOTA ESPECIAL:** Tenga en cuenta que la librería JFREECHART es de prueba gratuita en este caso, por lo que se le pide que porfavor ejecute el codigo desde el entorno de desarrollo puesto que el archivo .jar no corre por lo mencionado anteriormente, gracias por comprender.

Paso 1

Para ingresar los valores de la cantidad de hombres, mujeres, años, espacio deberá posesionarse en el apartado superior donde indica los datos de entrada, al frente de los ítems están los espacios donde tendrá que ingresar los datos que obligatoriamente se solicita que introduzca datos únicamente numéricos en especial de enteros ya que el software no recibirá otro tipo de dato.

CANTIDAD DE MUJERES		CANTIDAD DE HOMBRES	
TIEMPO EN AÑOS		TAMAÑO DEL ESPACIO	

Imagen N°16 Introducción de datos parametrizados

Paso 2

Al realizar el paso anterior por favor dirígete a la opción de "aceptar" uqe estará ubicado en la esquina superior derecha, con esta opción validará los datos y se ejecutará la simulación.

	ACEPTAR

Imagen N°17 opción de validación de la información

Paso 3

En el siguiente panel se mostrará la información ya validada y puesta en ejecución, aparte se calcularon la cantidad de días que transcurrieron a partir de los años ingresados, también nos muestra la cantidad de muertes por hombre y mujer que hubieron, por otra parte se visualiza la cantidad de agentes que surgieron o nacieron.

Poblacion: 1250	Intervalo de tiempo: 20 años	Intervalo de tiempo: 7300 días	Nacimientos: 295
Muertes de mujeres: 19	Muertes de hombres: 8	Tamaño del espacio: 100	

Imagen N°18 panel de información

Paso 4

En este apartado se visualiza los datos en gráficos, este panel se crea para experimentar el comportamiento de determinado número de agentes en un espacio determinado y unos parámetros ingresados.

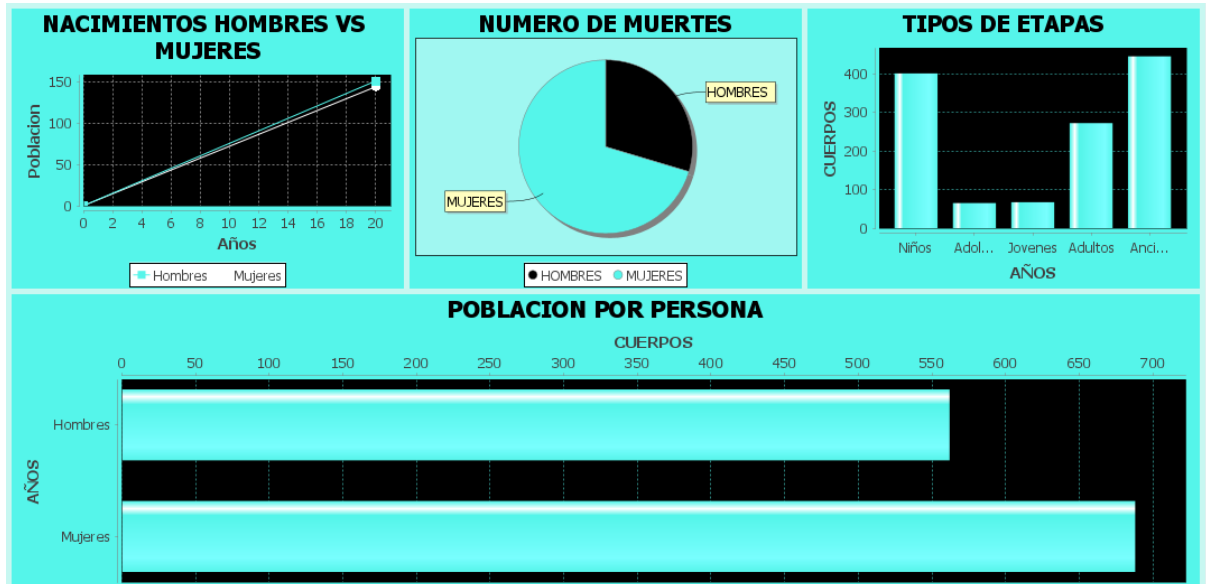


Imagen N°19 panel de información gráfica

Paso 5:

Disfrute de la simulación y sus resultados.