UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGIA, LA INDUSTRIA Y LOS RECURSO NATURALES NO RENOVABLES.

CARRERA DE INGENIERIA EN SISTEMAS

Simulación

Nombre:

Leonardo Vicente Paredes Rivas

Paralelo:

6ro "A" CIS

Docente:

Ing. Marlon Viñan

Loja–Ecuador
2019

Introducción

En el siguiente documento se va a describir el funcionamiento de lo generadores congruenciales lineales, específicamente el generador “Congruencial mixto” a través de un programa realizado en el lenguaje de programación Java.
Se explicara paso a paso en funcionamiento del programa así como el resultado que arroja.

Desarrollo

Generadores Congruenciales Lineales

Estos son capaces de generar una secuencia de números pseudoaleatorios en el cual el próximo número pseudoaleatorio es determinado a partir del último número generado, es decir, el número pseudoaleatorio X_{n+1} es derivado a partir del número pseudoaleatorio X_n

Para el caso particular del generador congruencial mixto, la relación de recurrencia es la siguiente $X_{n+1} = (a\,X_n + c)m$
Donde:

$X_{0=}$ es la semilla.
a = es el multiplicador.
c = es la constante aditiva.
m = es el modulo.

Funcionamiento del Programa del generador congruencial mixto

Para desarrollar el programa Congruencial mixto, se empieza creando clase que se va a llamar **“public class CongruencialMixto**”. Dentro de la clase, se declaran la variables que se van a utilizar para el funcionamiento del programa y almacenamiento de los resultados, en este caso las variables para la semilla (X0), el multiplicador (a), la constante aditiva (c), el modulo (m), el resultado del seudocódigo X_{n+1} (Xn1) y el resultado del seudocódigo X_n (xn).

```
public class CongruencialMixto {
    public static void main(String[] arg) {

        double a, c ;
        int xn=0,X0, m;
        double xn1=0;
        double nf=0;
```

A continuación es necesario importa la clase Scanner,

```
import java.util.Scanner;
```

esta Clase permitirá ingresar datos a través del teclado y almacenarlas en la variables anteriormente declaradas.

```
Scanner num = new Scanner(System.in);
System.out.println("Ingrese Constante Multiplicativa 'a'");
a = num.nextDouble();
System.out.println("Ingrese Costante Aditiva 'c'");
c = num.nextDouble();
System.out.println("Ingrese semilla 'X0'");
X0 = num.nextInt();
System.out.println("Ingrese modulo 'm'");
m = num.nextInt();
```

Una vez realizado los pasos anteriores, para que el programa funcione correctamente es necesario emplear un ciclo repetitivo, en este caso un ciclo for. Este permitirá que arroje una serie de resultado varias veces, hayas que se cumpla la condición que se le impuso en esta caso que inicie en 0 y sea menor a 8, es decir que imprima 7 veces los resultado deseados.

```
for(int i=0; i<8; i++){
```

Dentro del ciclo for se escribe el código que dará funcionamiento al programa. Como ya se declararon las variable anteriormente, ahora solo se procede al desarrollo de las formulas.

En **"xn1"** se almacena el resultado de la fórmula:

$$X_{n+1} = (a\,X_n + c)m$$

En **"xn"** se almacena el resultado del residuo de la formula

$$X_{n+1} = (a\,X_n + c)m$$

En **"nf"** se almacena el resultado de xn1/m (modulo).

```
xn1= ((a*X0)+c)/m;
xn= (int)((a*X0)+c)%m;
nf=(double)xn/m;
System.out.println(i+" | "+ X0 + "   |  "+ formato1.format(xn1)  +"    |  " +xn+ "     | "  +formato1.format(nf));
X0=xn;
```

Todo este ciclo repetitivo permitirá imprimir, mediante un System.out.println, los resultados deseados, es decir cuando a=5, c7, $X_{0=}$ 4, m=8, el programa arroja el siguiente resultado.

```
n | Xn   | (aXn+c)m   | Xn+1    |# Uniforme
0 | 4    |  3,38      |  3      | 0,38
1 | 3    |  2,75      |  6      | 0,75
2 | 6    |  4,62      |  5      | 0,62
3 | 5    |  4,00      |  0      | 0,00
4 | 0    |  0,88      |  7      | 0,88
5 | 7    |  5,25      |  2      | 0,25
6 | 2    |  2,12      |  1      | 0,12
7 | 1    |  1,50      |  4      | 0,50
BUILD SUCCESSFUL (total time: 3 seconds)
```