



**UNIVERSIDAD AUTÓNOMA METROPOLITANA  
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA  
UNIDAD AZCAPOTZALCO**

**Laboratorio de Análisis y simulación de sistemas**

**Práctica 1.**

***“Generación de números pseudoaleatorios”***

**Realizado por:**

- **GABRIEL FRANCISCO RAMOS 209302867**

**Profesor:**

**ANTONIN SEBASTIEN PONSICH**

---

**FECHA DE ENTREGA: 18 de octubre de 2011**

**TRIMESTRE: 11-Otoño**

**GRUPO: CSI02**

INDICE	PAG.
<b>PARTE 1</b> .....	<b>3</b>
1. <i>Código fuente (VBA)</i> .....	<b>3</b>
1.1. <i>Descripción del código fuente</i> .....	<b>4</b>
2. <i>Corridas (juego de parámetros)</i> .....	<b>5</b>
2.1 <i>Caso 1</i> .....	<b>5</b>
2.2 <i>Caso 2</i> .....	<b>6</b>
2.3 <i>Caso 3</i> .....	<b>7</b>
2.4 <i>Caso 4</i> .....	<b>8</b>
3. <i>Comentario de la eficacia de las cuatro instancias del generador</i> .....	<b>9</b>
 <b>PARTE 2</b> .....	 <b>9</b>
1. <i>Código fuente</i> .....	<b>9</b>
1.1 <i>Descripción del código fuente</i> .....	<b>11</b>
2. <i>Corrida (N=5000)</i> .....	<b>12</b>
3. <i>Calculo teórico de la media y desviación estándar</i> .....	<b>13</b>
3.1 <i>Comentario entre la diferencia del valor teórico y el valor simulado</i> .....	<b>13</b>
4. <i>Simulación variando N números aleatorios (500 a 10,000)</i> .....	<b>13</b>
4.1 <i>Conclusiones</i> .....	<b>13</b>

## PARTE 1:

### 1. Código fuente (VBA):

```
Option Explicit

Sub aleatorios()

'declaración de variables

Dim a As Double, c As Double, m As Double, x0 As Double

Dim x As Double, x1 As Double

Dim i As Integer, N As Integer, u() As Double

'pedir valores al usuario de cada uno de las variables

N = InputBox("¿Número aleatorios a generar?")

a = InputBox("¿valor de a?")

m = InputBox("¿valor de m?")

c = InputBox("¿valor de c?")

x0 = InputBox("¿valor de x0?")

ReDim u(N) As Double 'redimensionar el vector u

'hacer el siguiente ciclo a partir de 1 hasta N números de aleatorios a generar

For i = 1 To N

x = Int((a * x0 + c) / m) 'hacer la primera operación

x1 = (a * x0 + c) - x * m 'hacer la segunda operación utilizando el valor de x que salió en la primera operación.

u(i) = x1 / m

x0 = x1          'x0 toma el valor de x1 que salió de la operación anterior

Cells(i, 16).Value = u(i) 'se imprime todos los números aleatorios en la columna Pi

Next i

For i = 1 To 20

Cells(i, 1).Value = i 'se imprime una numeración de 1 al 20 desde A1 hasta A20

Cells(i, 2).Value = u(i) 'se imprime los primeros 20 números aleatorios desde B1 a B20

Next i

MsgBox ("¡La ejecución ha terminado exitosamente!") 'mensaje que le avisa al usuario que la ejecución a terminado

End Sub
```

### 1.1 Descripción del código fuente.

El programa anterior tiene como finalidad generar números aleatorios, en donde el usuario puede indicar el número de ellos e introducir los demás parámetros.

- ✓ En primer momento se declara todas variables que se va utilizar y además el tipo de datos de las variables, la mayoría de ellos se declaró como "Double" (doble precisión), excepto las variables "i" (contador) y "N". A continuación se hace un resumen de todas ellas.

Variabes	Tipo de variables	Uso
a	Double (doble precisión)	Parámetro solicitado (multiplicador)
c	Double	Parámetro solicitado (incremento)
m	Double	Parámetro solicitado (modulo)
x0	Double	Parámetro solicitado (semilla)
x	Double	Almacena temporalmente los datos
x1	Double	Almacena temporalmente los datos
i	Integer (entero)	Contador
N	Integer	Número de aleatorios a generar (solicitado)
u()	Double	Vector, Almacena los datos

- ✓ En segundo instante, el programa al momento de ejecutar, lo primero que hace es solicitar los valores (parámetros) que se ocupará durante la ejecución del mismo.
- ✓ Después se redimensiona el vector U () a u(N).
- ✓ A continuación se hace el ciclo (desde 1 hasta N números) que permite generar los "N" números aleatorios, utilizando todos los parámetros establecidos con anterioridad. Dentro del ciclo lo primero que se hace es utilizar la función "Int", con la intención de que nos devuelva (cada vez que repita el ciclo) la parte entera del número y después este valor depositarlo en la variable "x", una vez hecho esto se procede a hacer la segunda operación  $((a * x0 + c) - x * m)$  y el resultado de dicha operación se deposita en la variable "x1", teniendo este valor (x1) se divide entre "m" y este resultado (números aleatorios) se va depositando o imprimiendo en las celdas de la hoja de cálculo, en la columna "P". Nota. Cada vez que se repite el ciclo la variable "x0" toma el valor de "x1", con la intención de que los números aleatorios que se está generando sean diferentes.
- ✓ Después se imprime en la hoja de cálculo (Excel) una numeración del 1 al 20 en la columna A (A1-A20) y también los primeros 20 números aleatorios generados y que se muestra en la columna B(B1-B20).
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

## 2. Corridas (juego de parámetros).

Nota: en todos los casos el valor de la semilla  $x_0=245$  y  $N=10000$  números aleatorios.

### 2.1 Caso 1.

$a=214013$ ,  $c=2531011$  y  $m=2^{32}$

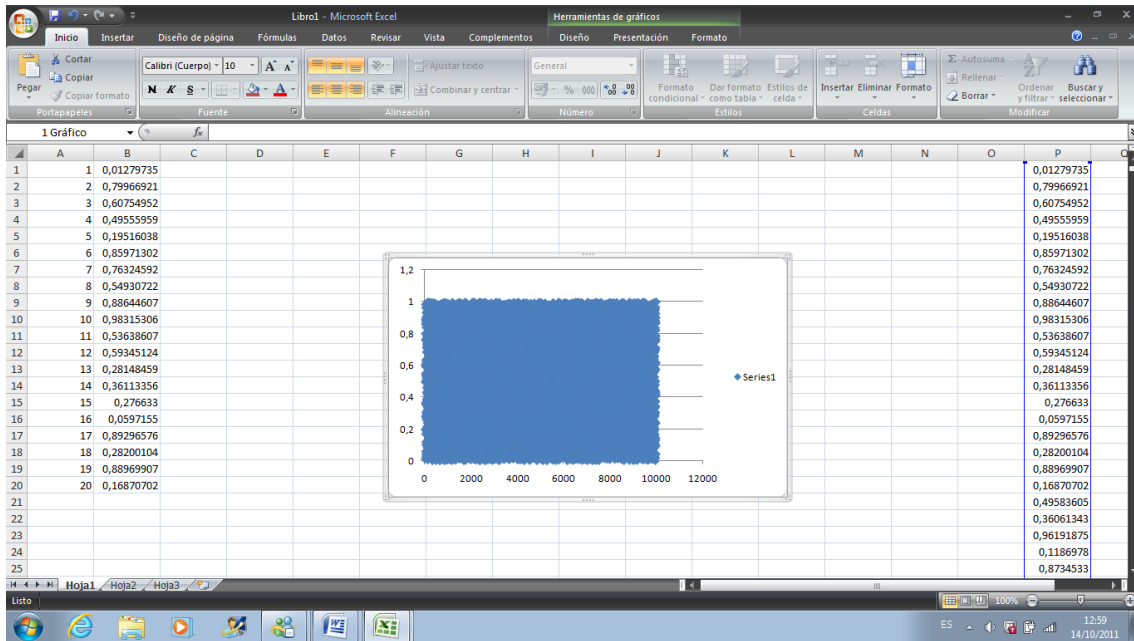
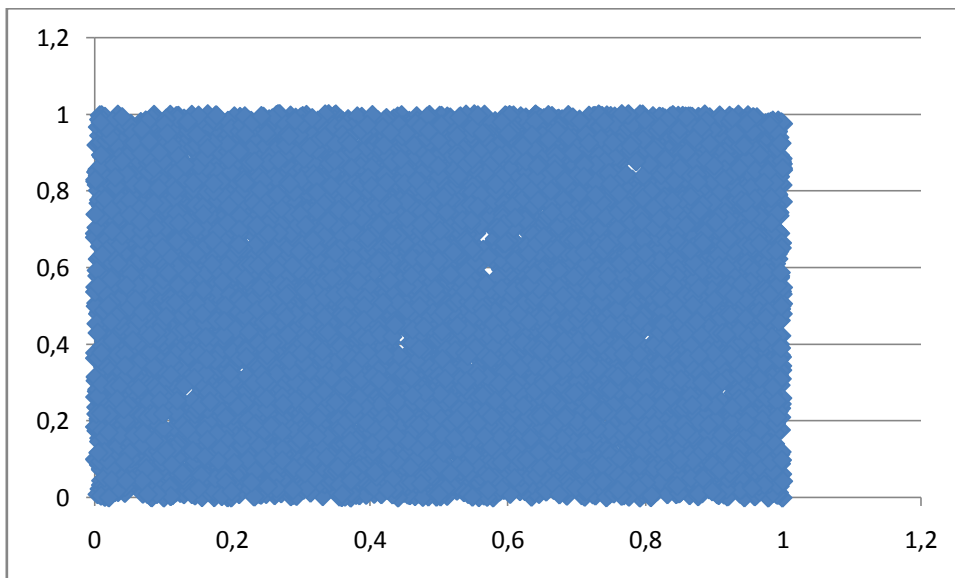


Figura 1: simulación realizada en VBA con los parámetros citados (Caso 1) y los números aleatorios se imprimen en la hoja de cálculo (Excel). Y la pequeña grafica que se muestra en la figura indica como están distribuidos los 10,000 números aleatorios generados.

Gráfica de  $u_{j+1} = f(u_j)$ .



## 2.2 Caso 2.

$a=65539$ ,  $c=0$  y  $m=2^{31}$

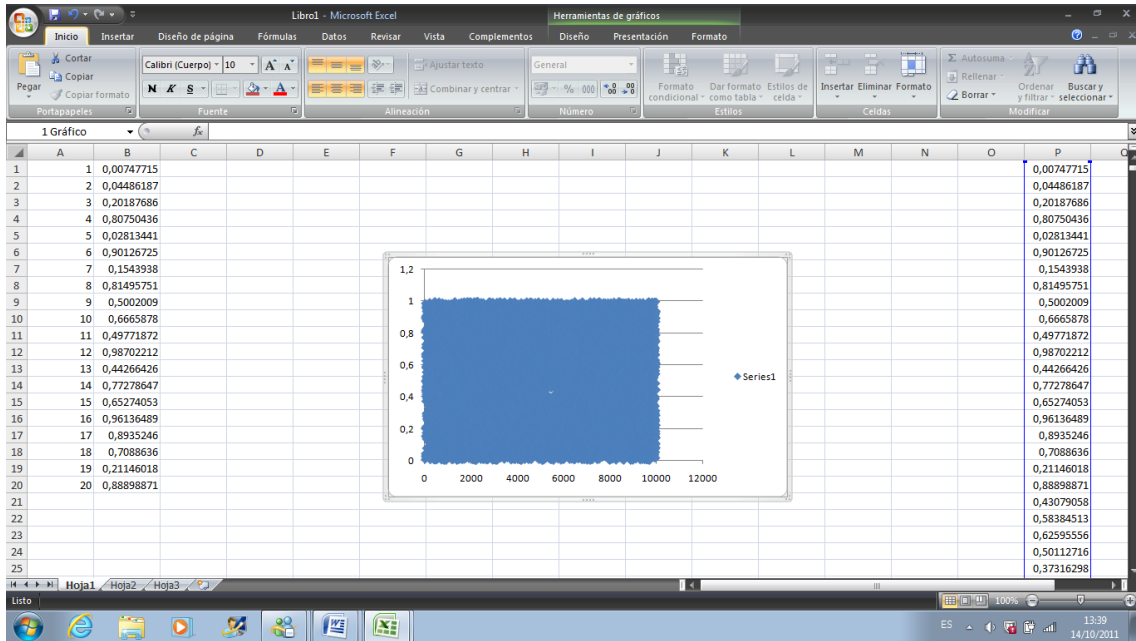
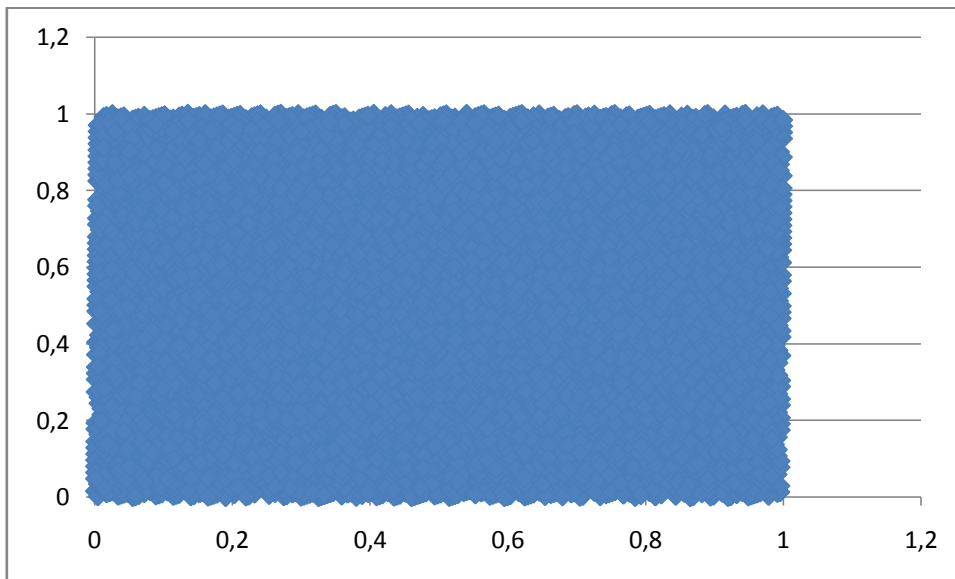


Figura 2: simulación realizada en VBA con los parámetros citados (Caso 2) y los números aleatorios se imprimen en la hoja de cálculo (Excel). Y la pequeña grafica que se muestra en la figura indica como están distribuidos los 10,000 números aleatorios generados.

Gráfica de  $u_{j+1} = f(u_j)$ .



### 2.3 Caso 3.

$a=237$ ,  $c=0$  y  $m=14657$

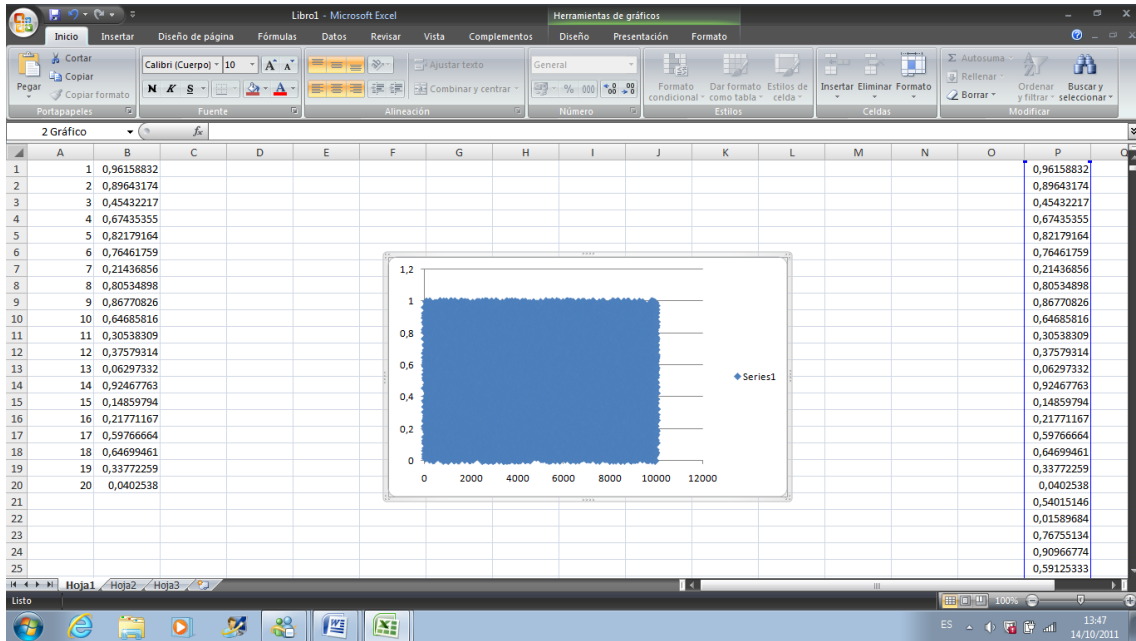
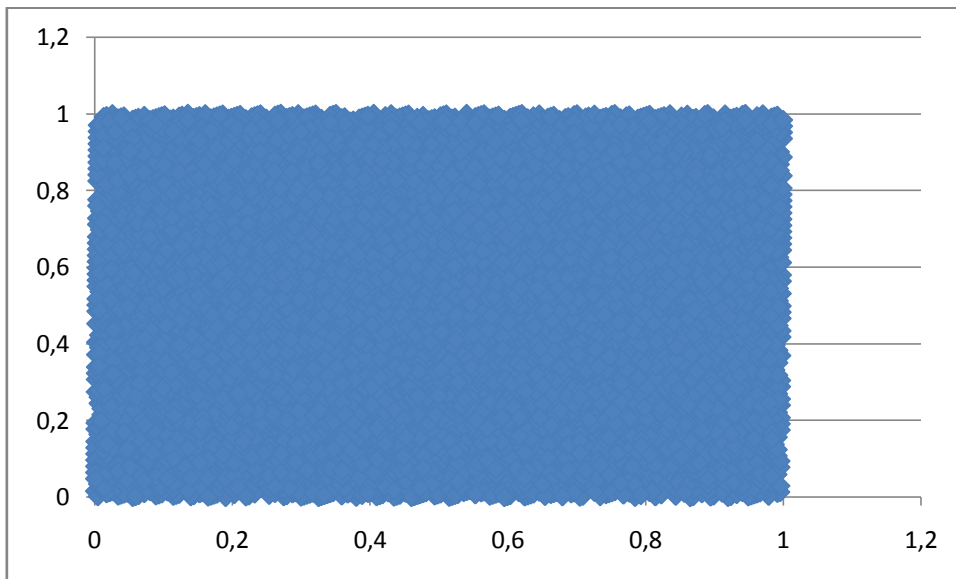


Figura 3: simulación realizada en VBA con los parámetros citados (Caso 3) y los números aleatorios se imprimen en la hoja de cálculo (Excel). Y la pequeña grafica que se muestra en la figura indica como están distribuidos los 10,000 números aleatorios generados.

Gráfica de  $u_{j+1} = f(u_j)$ .



## 2.4 Caso 4.

$a=23$ ,  $c=0$  y  $m=267$

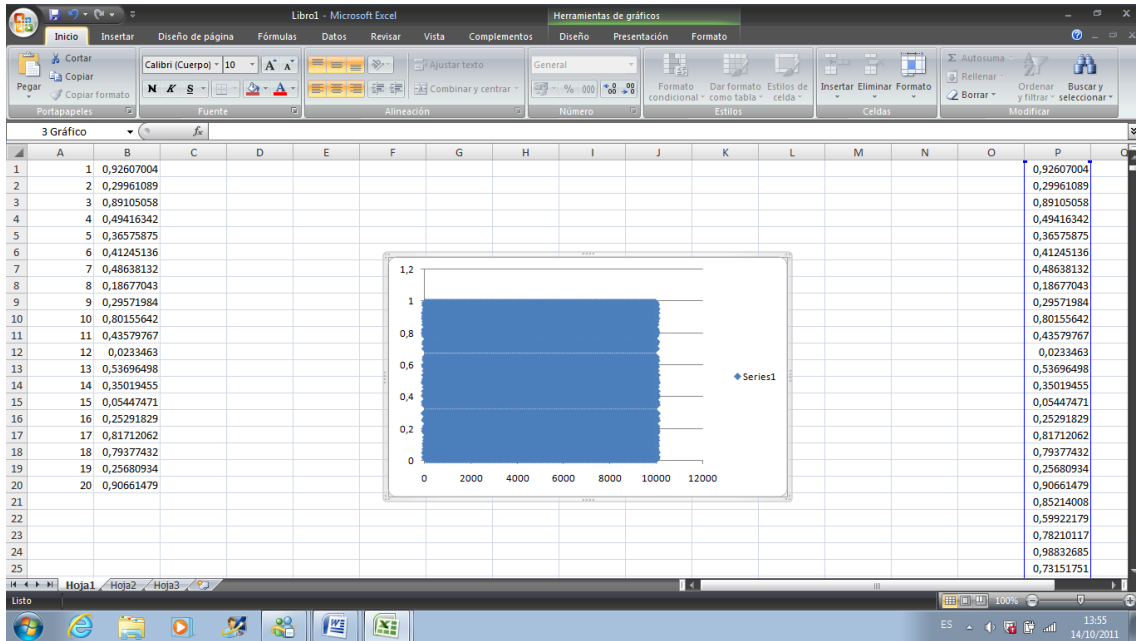
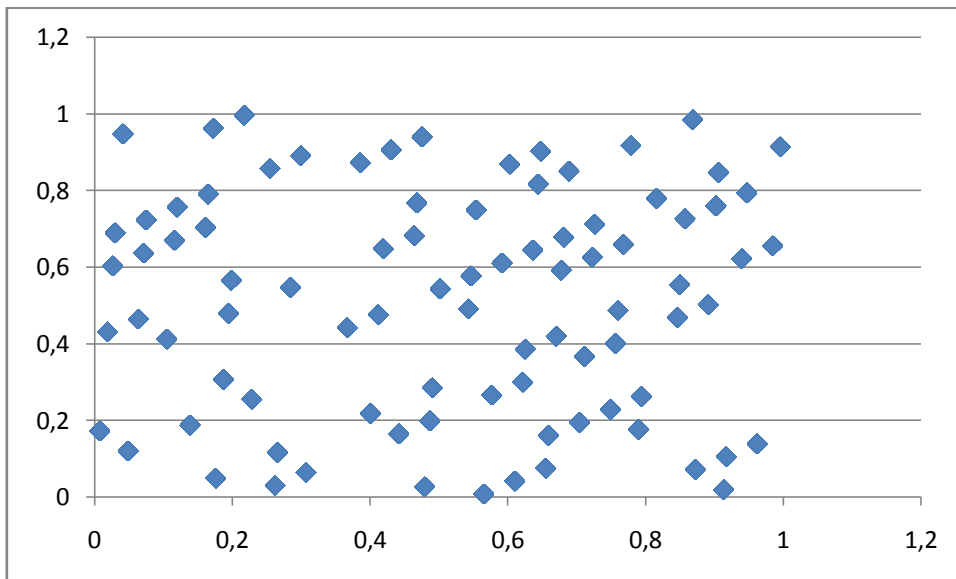


Figura 4: simulación realizada en VBA con los parámetros citados (Caso 4) y los números aleatorios se imprimen en la hoja de cálculo (Excel). Y la pequeña grafica que se muestra en la figura indica como están distribuidos los 10,000 números aleatorios generados.

Gráfica de  $u_{j+1} = f(u_j)$ .





### 3. Comentario de la eficacia de las cuatro instancias del generador.

Antes de comentar la eficacia de las 4 cuatro instancias (casos) del generador de números pseudoaleatorios, es importante recordar que los números aleatorios de esta práctica son producidos de manera determinista, sin embargo tienen toda la apariencia de ser variables o números aleatorias uniformes e independientes en el intervalo (0,1).

En el "caso 1" sostengo que tiene más eficacia que los demás, también podría ser el caso 2, sin embargo el más adecuado de estos cuatro casos viene siendo el caso 1 y el menos adecuado es sin duda el último caso (caso 4), esto es porque el número de variables (números aleatorios) que se puede generar antes de que comience la repetición tiene que ser grande (cuando un número empieza a repetirse toda la sucesión comienza a repetirse), entonces eso implica que el valor "m" debe de ser un número primo grande. Esa es la razón principal de que el caso 1 tiene más eficacia que los otros.

## PARTE 2:

### 1. Código fuente.

```
Option Explicit

Sub aleatorios()

'declaración de variables

Dim a As Double, c As Double, m As Double, x0 As Double

Dim z1 As Double, z2 As Double, suma As Double, media As Double

Dim var As Double, desviacion As Double

Dim x As Double, x1 As Double, varianza As Double

Dim i As Integer, N As Integer, u As Double, u1() As Double

'solicitud de valores al usuario

N = InputBox("Número aleatorios a generar")

z1 = InputBox("valor de Z1")

z2 = InputBox("Valor de z2")

'asignación de valores de las variables

a = 214013

c = 2531011

m = 2 ^ 32

x0 = 245

ReDim u1(N) As Double 'redimensionar al vector u

'ciclo que permite generar números aleatorios entre z1=2.38 a z2=2.64
```

```

For i = 1 To N

x = Int((a * x0 + c) / m)

x1 = (a * x0 + c) - x * m

u = x1 / m

u1(i) = (z1) + u * (z2 - z1)

x0 = x1

Cells(i, 16).Value = u1(i) 'los números aleatorios se imprime en la columna P

Next i

suma = 0 'ciclo que permite calcular la media de los N números

For i = 1 To N

suma = suma + u1(i)

Next i

media = suma / N

Cells(2, 4).Value = "media"

Cells(3, 4).Value = media

var = 0 'ciclo que permite calcular la desviación estándar

For i = 1 To N

var = var + (u1(i) - media) ^ 2

Next i

varianza = var / (N - 1)

desviacion = Sqr(varianza)

Cells(2, 5).Value = "desv. estandar"

Cells(3, 5).Value = desviacion

For i = 1 To 20

Cells(i, 1).Value = i 'imprime una numeración del 1 al 20 en la columna A

Cells(i, 2).Value = u1(i) 'imprime los primeros 20 números aleatorios

Next i

MsgBox ("¡La ejecución ha terminado exitosamente!")

End Sub

```

### 1.1 Descripción del código fuente.

Este programa tiene como finalidad generar N números aleatorios entre ciertas cotas de distribución uniforme que el usuario puede indicar y los demás parámetros del generador están declarados y definidos dentro del código. Nota (las cotas para este generador son  $z1=2,38$  y  $z2=2,64$ ).

- ✓ En primer momento se declara todas variables que se va utilizar y además el tipo de datos de las variables, la mayoría de ellos se declaró como “Double” (doble precisión), excepto las variables “i” (contador) y “N”. A continuación se hace un resumen de todas ellas.

Variabes	Tipo de variables	Uso
a	Double (doble precisión)	Parámetro declarado (multiplicador)
c	Double	Parámetro declarado (incremento)
m	Double	Parámetro declarado (modulo)
x0	Double	Parámetro declarado (semilla)
z1	Double	Cota inicial solicitado
z2	Double	Cota final solicitado
suma	Double	Almacena temporalmente los datos
media	Double	Almacena temporalmente los datos
var	Double	Almacena temporalmente los datos
desviacion	Double	Almacena temporalmente los datos
x	Double	Almacena temporalmente los datos
x1	Double	Almacena temporalmente los datos
varianza	Double	Almacena temporalmente los datos
i	Integer (entero)	Contador
N	integer	Número de aleatorios a generar (solicitado)
u	Double	Almacena temporalmente los datos
u1()	Double	Vector, almacena los datos

- ✓ En segundo instante, el programa al momento de ejecutar, lo primero que hace es solicitar el número de aleatorios a generar (N) y las cotas (z1 y z2).
- ✓ Después se redimensiona el vector U1 () a u1(N).
- ✓ A continuación se hace el ciclo (desde 1 hasta N números) que permite generar los “N” números aleatorios, utilizando todos los parámetros establecidos con anterioridad. Dentro del ciclo lo primero que se hace es utilizar la función “Int”, con la intención de que nos devuelva (cada vez que repita el ciclo) la parte entera del número y después este valor depositarlo en la variable “x”, una vez hecho esto se procede a hacer la segunda operación  $((a * x0 + c) - x * m)$  y el resultado de dicha operación se deposita en la variable “x1”, teniendo este valor (x1) se divide entre “m” y nuevamente este valor se deposita en la variable(u). Posteriormente se hace otra operación  $(u1(i) = (z1) + u * (z2 - z1))$ , con el objetivo de que los números aleatorios que se está generando estén dentro de las cotas o intervalos (2,38 – 2,64) y por último estos números aleatorios se imprime en la columna P. Nota. Cada vez que se repite el ciclo la variable “x0” toma el valor de “x1”, con la intención de que los números aleatorios que se está generando sean diferentes.

- ✓ Posteriormente se procede a calcular la media. Para ello se hace un ciclo (de 1 hasta N) y dentro de este ciclo se calcula la suma total de los N números aleatorios y se deposita en la variable (suma), y una vez teniendo la suma total, afuera del ciclo se divide entre los N números aleatorios generados y posteriormente el resultado de esta operación (media) se imprime en la celda (“D3”) de la hoja de datos (Excel).
- ✓ Para calcular la desviación estándar, también se hace un ciclo (1 hasta N), y dentro del ciclo se hace la siguiente operación ( $var = var + (u1(i) - media)^2$ ) hasta tener el total de los N números y se deposita en la variable (var) y afuera del ciclo este valor se divide entre “N-1” y este resultado se deposita en la variable “varianza” y después se calcula la raíz cuadrada del valor que se encuentra en la “varianza” y el resultado viene siendo la “desviación estándar” y por ultimo este valor se imprime en la celda (“E3”) de la hoja de datos (Excel).
- ✓ Después se imprime en la hoja de cálculo (Excel) una numeración del 1 al 20 en la columna A (A1-A20) y también los primeros 20 números aleatorios generados y que se muestra en la columna B (B1-B20).
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

## 2. Corrida (N=5000)

$z1=2,38$  y  $z2=2,64$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		1	2,3832731														2,3832731
2		2	2,58791399		media	desv. estandar											2,58791399
3		3	2,53796287		2,50930857	0,0758916											2,53796287
4		4	2,50884549														2,50884549
5		5	2,4307417														2,4307417
6		6	2,60352539														2,60352539
7		7	2,57844394														2,57844394
8		8	2,52281988														2,52281988
9		9	2,61047598														2,61047598
10		10	2,6356198														2,6356198
11		11	2,51946038														2,51946038
12		12	2,53429732														2,53429732
13		13	2,45318599														2,45318599
14		14	2,47389472														2,47389472
15		15	2,45192458														2,45192458
16		16	2,39552603														2,39552603
17		17	2,6121711														2,6121711
18		18	2,4532027														2,4532027
19		19	2,61132176														2,61132176
20		20	2,42386382														2,42386382
21																	2,50891737
22																	2,47375949
23																	2,63009888
24																	2,41086143
25																	2,60709786

Figura 5: simulación realizada en VBA con los parámetros citados y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor de la media es 2.5093 y la desviación estándar es 0.07589

### 3. Cálculo teórico de la media y desviación estándar.

Formulas:  $x^- = \frac{z1+z2}{2}$  donde:  $x^-$  =media; (z1 y z2)=cotas

$\sigma = \sqrt{\frac{(z2-z1)^2}{12}}$  donde:  $\sigma$  =desviación estándar; (z1 y z2)=cotas

Cálculo teórico de la media:

$$x^- = \frac{2.38+2.64}{2} = 2.51$$

Cálculo teórico de la desviación estándar:

$$\sigma = \sqrt{\frac{(2.64-2.38)^2}{12}} = 0.075055$$

#### 3.1 Comentario entre la diferencia del valor teórico y el valor simulado.

Para los 5000 números aleatorios generados la diferencia entre el valor simulado (media y desviación estándar) y el valor teórico es muy pequeña, el porcentaje de error relativo es menor al 1.15%, entonces se puede aceptar los valores simulados.

### 4. Simulación variando N números aleatorios (500 a 10,000).

N (números aleatorios)	Media (simulado)	Desviación estándar (simulado)	Error relativo porcentual (Ep) para la media	Error relativo porcentual (Ep) para la desviación estándar
500	2.51332791	0.07758806	0.13258 %	3.37493%
2,000	2.5095785	0.07553227	0.01679%	0.63589%
4,000	2.50908422	0.07592285	0.03648%	1.15628%
7,000	2.50978128	0.07539636	0.00871%	0.45481%
10,000	2.50991643	0.07519163	0.00332%	0.18203%

La tabla anterior muestra las 5 corridas con sus respectivos valores de la media y desviación estándar, variando el valor de N. **Nota:** El error relativo porcentual se calcula de la siguiente manera.  $Ep = \frac{Valor\ teórico - |Valor\ simulado|}{Valor\ teórico} (100\%)$

#### 4.1 Conclusiones.

Para poder escribir una conclusión, tuve la idea de calcular el error relativo porcentual y a través de ello observar que tan cercano se encuentran los valores simulados con respecto a los valores teóricos, lo interesante de esto es que se puede observar que cuando simulamos con N=2000 números aleatorios hay una disminución interesante de error relativo porcentual (Ep) tanto con la media y como la desviación estándar, pasamos con N=4000 números aleatorios vuelve a incrementarse el Ep y a partir de ese número el Ep se va disminuyendo conforme aumentamos el valor de N. Habría que estudiar un poco más a fondo que sucede en el intervalo N=2000 a N=4000 números aleatorios, pero por lo pronto, para hacer una conclusión expreso nos quedamos que entre más grande sea N, los valores simulados estarán más cercanos de los valores teóricos.