UNIVERSIDAD AUTÓNOMA METROPOLITANA
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA
UNIDAD AZCAPOTZALCO

Laboratorio de Análisis y simulación de sistemas

Práctica 2.

Realizado por:

- **GABRIEL FRANCISCO RAMOS 209302867**

Profesor:

ANTONIN SEBASTIEN PONSICH

FECHA DE ENTREGA: 25 de octubre de 2011

TRIMESTRE: 11-Otoño

GRUPO: CSI02

INDICE **PAG.**

PRUEBA DE JI CUADRADA

1. Código fuente(VBA).

```
Option Explicit

Sub aleatorios()

'declaración de variables

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

Dim x As Double, x1 As Double, k As Integer, ciclo As Boolean

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

Dim intinf As Double, intsup As Double, cuadra() As Double, cuadrato As Double

'pedir valores al usuario de cada uno de las variables

N = InputBox("numero de pseudoaleatorios a generar")

a = InputBox("valor de a")

m = InputBox("valor de m")

c = InputBox("valor de c")

x0 = InputBox("valor de x0")

k = InputBox("numero de  clases")

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

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

For i = 1 To N

x = Int((a * x0 + c) / m) 'realizar la siguiente operación

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

u(i) = x1 / m

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

Cells(i + 1, 1).Value = u(i) 'se imprime todos los resultados la columna A1

Next i
```

```
ReDim conta(N) As Integer 'redimensionar el vector conta

m = N / k 'numero de datos esperados en la clase

'ciclo que permite encontrar la frecuencia en cada clase

For i = 1 To N

j = 1

ciclo = True

Do While (j <= k) And (ciclo = True)

ciclo = False

If u(i) >= (j - 1) * 0.02 And u(i) < j * 0.02 Then

conta(j) = conta(j) + 1

Cells(j + 1, 5).Value = conta(j)

Cells(1, 5).Value = "Frecuencia"

End If

j = j + 1

ciclo = True

Loop

Next i

anchoclase = 1 / k 'ancho en cada clase

'ciclo que permite imprimir en la hoja , todas las clases

intinf = 0

For i = 1 To k

intinf = intinf + 0

intsup = intinf + anchoclase

Cells(1, 3).Value = "Intervalo Inf"

Cells(i + 1, 3).Value = intinf

Cells(1, 4).Value = "Intervalo Sup"
```

```
Cells(i + 1, 4).Value = intsup

intinf = intsup

Next i

m = N / k 'numero de datos esperados en cada clase

ReDim cuadra(k) As Double 'redimensionar el vector cuadra

'ciclo que permite calcular la ji cuadrada en cada clase

For i = 1 To k

cuadra(i) = (conta(i) - m) ^ 2 / m

Cells(i + 1, 7).Value = cuadra(i)

Next i

'ciclo que permite calcular la ji cuadrada total

cuadrato = 0

For i = 1 To k

cuadrato = cuadrato + cuadra(i)

Next i

Cells(2, 9).Value = cuadrato

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

End Sub
```

1.1 Descripción del código fuente:

El programa que se muestra en la tabla, tiene como finalidad agrupar en K clases los números aleatorios generados en la práctica 1 y calcular la frecuencia existente en cada clase para posteriormente calcular el estadístico de la ji cuadrada e imprimir el valor, en la hoja de cálculo (Excel).

Nota. En la parte de generación de los N números aleatorios, el procedimiento es exactamente lo mismo al de la práctica 1.

A continuación se hace una lista de todas las variables que se utilizó en la elaboración del código fuente.

Variables	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
anchoclase	Double	Almacena un valor constante
K	Integer	Numero de clases (valor cte.)
ciclo	Boolean	Condición
J	Integer	Contador
conta()	Integer	Contador
intinf	Double	Límite inferior de la clase
intsup	Double	Límite superior de la clase
cuadra()	Double	Vector (almacena datos)
cuadrato	Double	Almacena datos

- ✓ Declaración de variables.
- ✓ Solicitud de valores (parámetros) necesarias durante la ejecución de programa.
- ✓ Generación de N números aleatorios (lo mismo que se describe en la práctica 1)
- ✓ Obtención de la frecuencia en cada clase: se hace un ciclo que nos permita recorrer los N números aleatorios generados y a la vez, buscar en que clase se encuentra dichos números aleatorios. Para esto cada clase tiene que tener un límite inferior y un límite superior. Nota. En el código fuente se puede observar como está estructurado el ciclo. En primer instante se introduce un "for", que permita recorrer los N números aleatorios, dentro del " for" se introduce un " Do while", esta sentencia permite condicionar de que se tiene que recorrer hasta K clases (no puede ser más) y dentro de esta sentencia se añade un "if", cuya sentencia permite declarar el limite inferior y superior de cada clase. En general este ciclo permite encontrar cuantos números aleatorios (frecuencia) se encuentra en cada clase y posteriormente imprimir en la columna (E) de la hoja de cálculo (Excel).
- ✓ Se crea otro ciclo que permite generar los límites inferiores y superiores de cada clase. En el primero ciclo el límite inferior vale cero, mientras que el límite superior es igual al límite inferior más 0.02 (1/k). Se imprime las clases con sus respectivos límites (superior e inferior) y después el límite inferior tiene que tomar el valor del límite

superior para que en las demás clases siga la misma secuencia hasta generar las K clases e imprimirlos en la hoja de cálculo.

- ✓ Calculo estadístico de Ji cuadrada. En primer instante se crea un ciclo que permita recorrer la K clases e ir calculando en cada clase el valor estadístico de la ji cuadrada y estos valores se imprime en la columna (G). después se crea otro ciclo que nos permita calcular la ji cuadrada total, para esto se tiene que recorrer la ji cuadrada de cada clase e irlos sumando hasta obtener la ji cuadrada total.
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

1.2 Valor calculado: para un error α=5%, sobre la uniformidad de la serie de números producidos.

Grados libertad:

V=k-1=50-1=49 K=numero de clases; X^2=ji cuadrada.

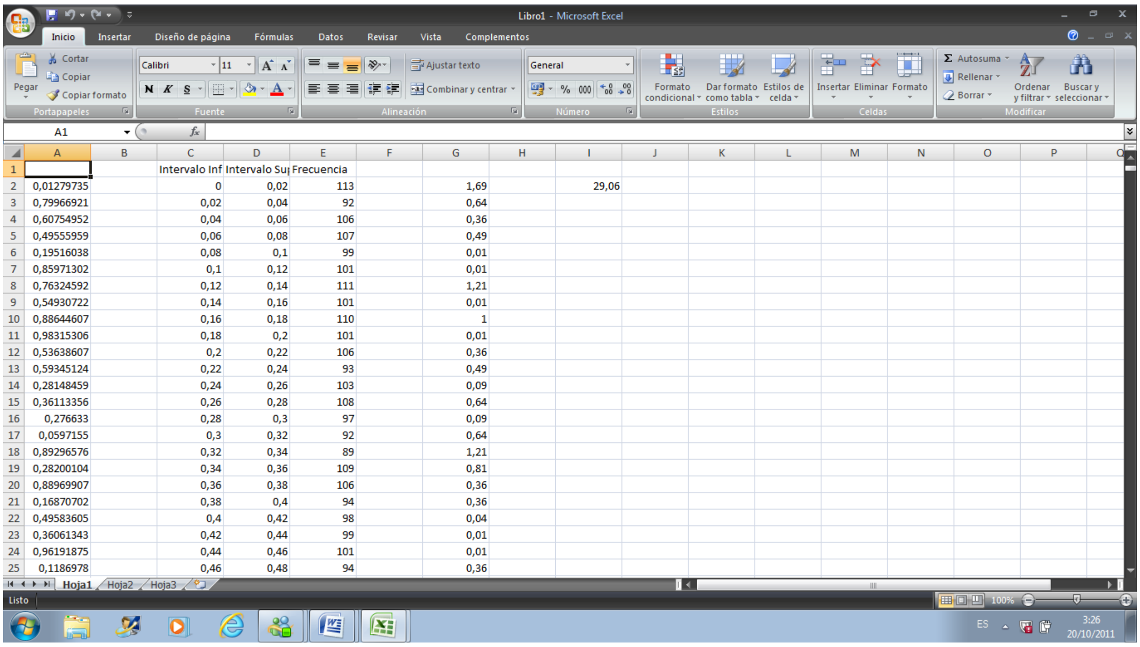
X^2(5%, 49)= 66.2538 (se obtuvo en tablas)

2. Juego de parámetros

Nota: en todos los casos el valor de la semilla x0=245, N=5,000 números aleatorios y k=50 clases

2.1 Caso 1

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



	A	B	C	D	E	F	G	H	I
1			Intervalo Inf	Intervalo Su|	Frecuencia				
2	0,01279735		0	0,02	113		1,69		29,06
3	0,79966921		0,02	0,04	92		0,64		
4	0,60754952		0,04	0,06	106		0,36		
5	0,49555959		0,06	0,08	107		0,49		
6	0,19516038		0,08	0,1	99		0,01		
7	0,85971302		0,1	0,12	101		0,01		
8	0,76324592		0,12	0,14	111		1,21		
9	0,54930722		0,14	0,16	101		0,01		
10	0,88644607		0,16	0,18	110		1		
11	0,98315306		0,18	0,2	101		0,01		
12	0,53638607		0,2	0,22	106		0,36		
13	0,59345124		0,22	0,24	93		0,49		
14	0,28148459		0,24	0,26	103		0,09		
15	0,36113356		0,26	0,28	108		0,64		
16	0,276633		0,28	0,3	97		0,09		
17	0,0597155		0,3	0,32	92		0,64		
18	0,89296576		0,32	0,34	89		1,21		
19	0,28200104		0,34	0,36	109		0,81		
20	0,88969907		0,36	0,38	106		0,36		
21	0,16870702		0,38	0,4	94		0,36		
22	0,49583605		0,4	0,42	98		0,04		
23	0,36061343		0,42	0,44	99		0,01		
24	0,96191875		0,44	0,46	101		0,01		
25	0,1186978		0,46	0,48	94		0,36		

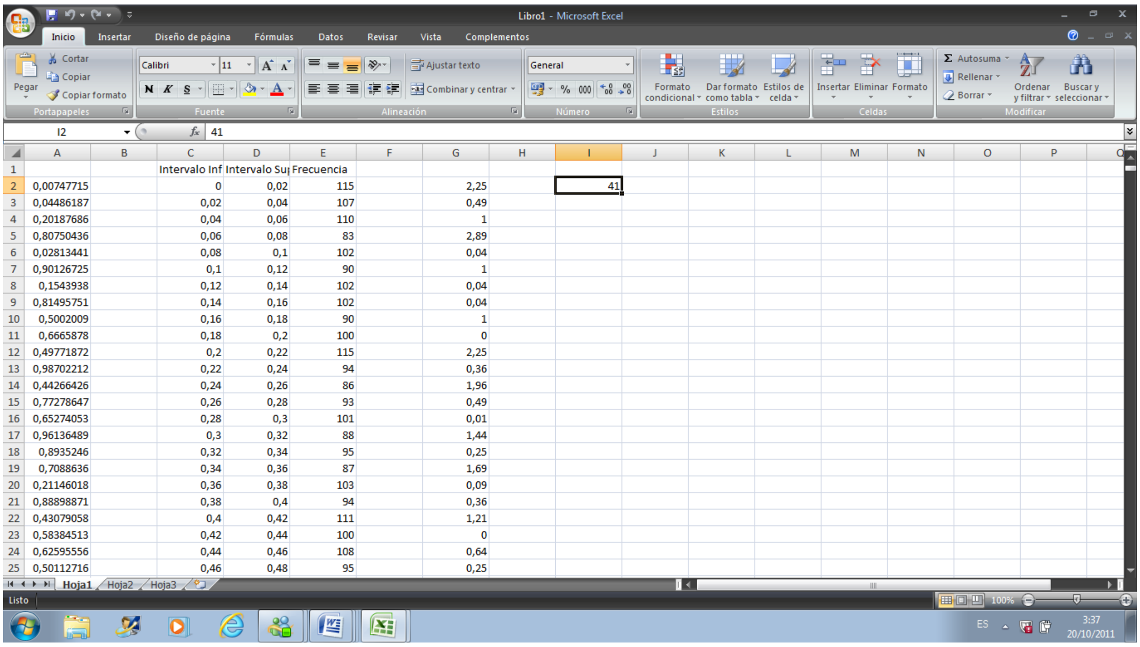
Figura 1: simulación realizada en VBA con los parámetros citados (Caso 1) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor de la ji cuadrada es 29.06

Entonces. X^2= 29.06

29.06 < 66.2538 Por lo tanto: "No se rechaza la hipótesis de uniformidad"

2.2 Caso 2

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



	A	B	C	D	E	F	G	H	I
1			Intervalo Inf	Intervalo Su	Frecuencia				
2	0,00747715		0	0,02	115		2,25		41
3	0,04486187		0,02	0,04	107		0,49		
4	0,20187686		0,04	0,06	110		1		
5	0,80750436		0,06	0,08	83		2,89		
6	0,02813441		0,08	0,1	102		0,04		
7	0,90126725		0,1	0,12	90		1		
8	0,1543938		0,12	0,14	102		0,04		
9	0,81495751		0,14	0,16	102		0,04		
10	0,5002009		0,16	0,18	90		1		
11	0,6665878		0,18	0,2	100		0		
12	0,49771872		0,2	0,22	115		2,25		
13	0,98702212		0,22	0,24	94		0,36		
14	0,44266426		0,24	0,26	86		1,96		
15	0,77278647		0,26	0,28	93		0,49		
16	0,65274053		0,28	0,3	101		0,01		
17	0,96136489		0,3	0,32	88		1,44		
18	0,8935246		0,32	0,34	95		0,25		
19	0,7088636		0,34	0,36	87		1,69		
20	0,21146018		0,36	0,38	103		0,09		
21	0,88898871		0,38	0,4	94		0,36		
22	0,43079058		0,4	0,42	111		1,21		
23	0,58384513		0,42	0,44	100		0		
24	0,62595556		0,44	0,46	108		0,64		
25	0,50112716		0,46	0,48	95		0,25		

Figura 2: simulación realizada en VBA con los parámetros citados (Caso 2) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor de la ji cuadrada es 41

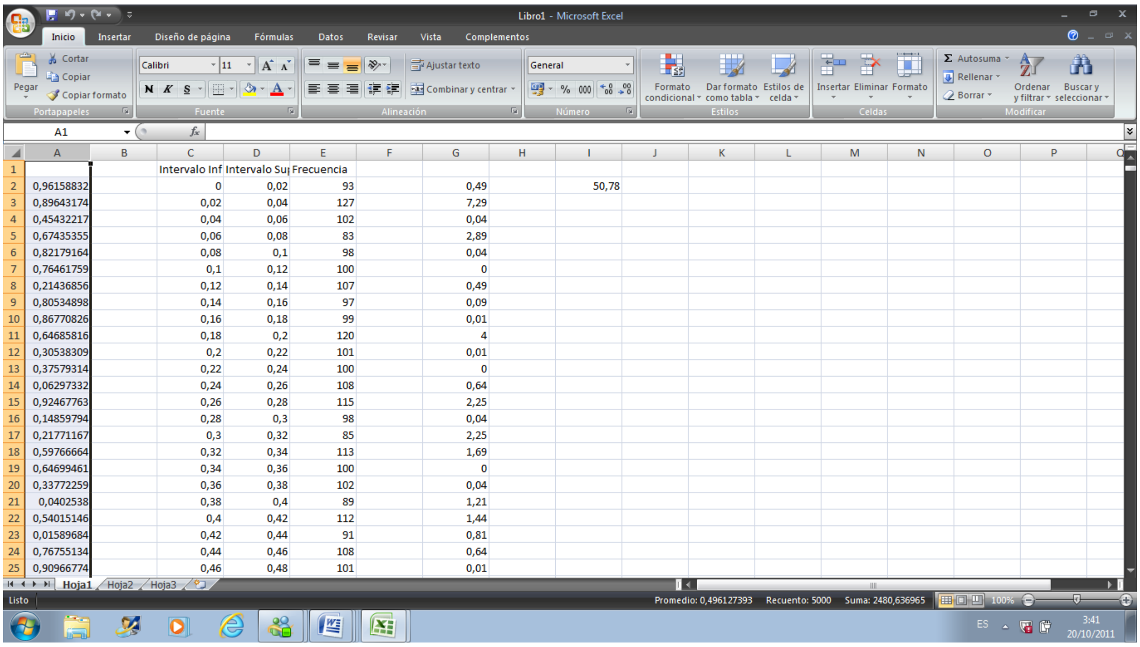
Entonces. X^2= 41

41 < 66.2538 Por lo tanto: "No se rechaza la hipótesis de uniformidad"

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). El valor de la ji cuadrada es 50.78



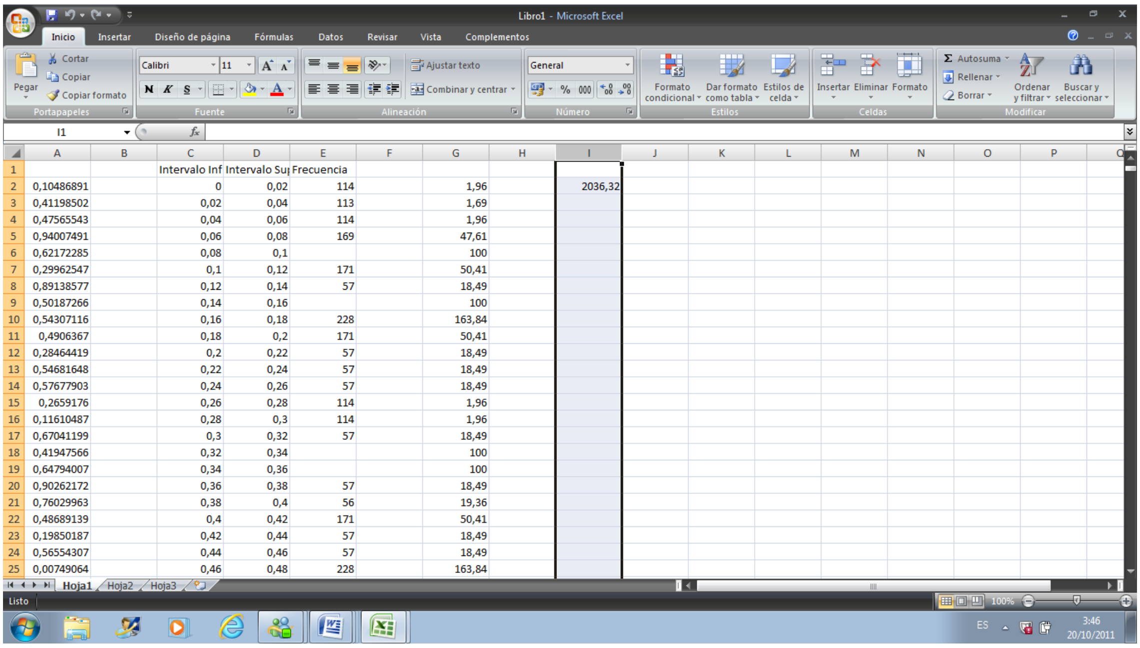
	A	B	C	D	E	F	G	H	I
1			Intervalo Inf	Intervalo Su	Frecuencia				
2	0,96158832		0	0,02	93		0,49		50,78
3	0,89643174		0,02	0,04	127		7,29		
4	0,45432217		0,04	0,06	102		0,04		
5	0,67435355		0,06	0,08	83		2,89		
6	0,82179164		0,08	0,1	98		0,04		
7	0,76461759		0,1	0,12	100		0		
8	0,21436856		0,12	0,14	107		0,49		
9	0,80534898		0,14	0,16	97		0,09		
10	0,86770826		0,16	0,18	99		0,01		
11	0,64685816		0,18	0,2	120		4		
12	0,30538309		0,2	0,22	101		0,01		
13	0,37579314		0,22	0,24	100		0		
14	0,06297332		0,24	0,26	108		0,64		
15	0,92467763		0,26	0,28	115		2,25		
16	0,14859794		0,28	0,3	98		0,04		
17	0,21771167		0,3	0,32	85		2,25		
18	0,59766664		0,32	0,34	113		1,69		
19	0,64699461		0,34	0,36	100		0		
20	0,33772259		0,36	0,38	102		0,04		
21	0,0402538		0,38	0,4	89		1,21		
22	0,54015146		0,4	0,42	112		1,44		
23	0,01589684		0,42	0,44	91		0,81		
24	0,76755134		0,44	0,46	108		0,64		
25	0,90966774		0,46	0,48	101		0,01		

Entonces. $X^2 = 50.78$

$50.78 < 66.2538$ Por lo tanto: “No se rechaza la hipótesis de uniformidad”

2.4 Caso 4

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



	A	B	C	D	E	F	G	H	I
1			Intervalo Inf	Intervalo Su	Frecuencia				
2	0,10486891		0	0,02	114		1,96		2036,32
3	0,41198502		0,02	0,04	113		1,69		
4	0,47565543		0,04	0,06	114		1,96		
5	0,94007491		0,06	0,08	169		47,61		
6	0,62172285		0,08	0,1			100		
7	0,29962547		0,1	0,12	171		50,41		
8	0,89138577		0,12	0,14	57		18,49		
9	0,50187266		0,14	0,16			100		
10	0,54307116		0,16	0,18	228		163,84		
11	0,4906367		0,18	0,2	171		50,41		
12	0,28464419		0,2	0,22	57		18,49		
13	0,54681648		0,22	0,24	57		18,49		
14	0,57677903		0,24	0,26	57		18,49		
15	0,2659176		0,26	0,28	114		1,96		
16	0,11610487		0,28	0,3	114		1,96		
17	0,67041199		0,3	0,32	57		18,49		
18	0,41947566		0,32	0,34			100		
19	0,64794007		0,34	0,36			100		
20	0,90262172		0,36	0,38	57		18,49		
21	0,76029963		0,38	0,4	56		19,36		
22	0,48689139		0,4	0,42	171		50,41		
23	0,19850187		0,42	0,44	57		18,49		
24	0,56554307		0,44	0,46	57		18,49		
25	0,00749064		0,46	0,48	228		163,84		

Figura 4: simulación realizada en VBA con los parámetros citados (Caso 4) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor de la ji cuadrada es 2036,32

Entonces. X^2= 2036,32

2036,32 > 66.2538 Por lo tanto: "Se rechaza la hipótesis de uniformidad"

3. Conclusión de los 4 casos (rechazo o no de la hipótesis de uniformidad).

En los tres primeros casos se acepta la hipótesis de uniformidad y el caso 4, es el único caso que se rechaza la hipótesis de uniformidad, para poder aceptar una hipótesis de uniformidad se requiere que el valor de m sea lo más grande posible, nos damos cuenta que no importa cuántos números aleatorios generemos, para poder tener una distribución uniforme lo importante es el valor que le demos a "m". Es por esa razón que en los tres primeros casos se acepta la hipótesis por que los valores de m son aceptables, mientras que el último caso, el valor de m es muy pequeño.

PRUEBA DE KOLMOGOROV-SMIRNOV

4. Código fuente.

```
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, sorteado As Boolean, aux As Double

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

Dim dmas() As Double, dmenos() As Double

'pedir valores al usuario de cada uno de las variables

N = InputBox("numero de pseudoaleatorios 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 apartir de 1 hasta N numeros de pseudoaleatorios a generar

For i = 1 To N

x = Int((a * x0 + c) / m) 'realizar la siguiente operación

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

u(i) = x1 / m

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

Cells(i + 1, 1).Value = u(i) 'se imprime todos los resultados la columna A1

Next i

i = 1

 sorteado = False ' ciclo que permite ordenar los numeros aleatorios

 Do While (i <= N - 1) And (sorteado = False)

 sorteado = True

 For j = 1 To N - i

    If u(j + 1) < u(j) Then

    aux = u(j)

    u(j) = u(j + 1)
```

```
    u(j + 1) = aux
    sorteado = False
    End If
 Next j
    i = i + 1
    Loop
For i = 1 To N 'imprime todos los datos ordenados en la columna B
Cells(1, 2).Value = "Datos ordenados"
Cells(i + 1, 2).Value = u(i)
Next i
ReDim dmas(N) As Double
ReDim dmenos(N) As Double
For i = 1 To N 'ciclo que calcula el D+  y el D- para cada numero aleatorio ordenado
dmas(i) = (i / N) - u(i)
dmenos(i) = u(i) - (i - 1) / N
Cells(1, 3).Value = "D +"
Cells(i + 1, 3).Value = dmas(i)
Cells(1, 4).Value = "D -"
Cells(i + 1, 4).Value = dmenos(i)
Next i
i = 1
 sorteado = False 'orden todos los D+ de mayor a menor
 Do While (i <= N - 1) And (sorteado = False)
 sorteado = True
 For j = 1 To N - i
    If dmas(j + 1) < dmas(j) Then
    aux = dmas(j + 1)
    dmas(j + 1) = dmas(j)
```

```
 dmas(j) = aux
    sorteado = False
    End If
 Next j
    i = i + 1
    Loop
Cells(1, 5).Value = "D + máx"
For i = N To N 'imprime el D+ mas grande
Cells(2, 5).Value = dmas(i)
Next i
 i = 1
 sorteado = False 'orden todos los D- de mayor a menor
 Do While (i <= N - 1) And (sorteado = False)
 sorteado = True
 For j = 1 To N - i
    If dmenos(j + 1) < dmenos(j) Then
    aux = dmenos(j + 1)
    dmenos(j + 1) = dmenos(j)
    dmenos(j) = aux
    sorteado = False
    End If
 Next j
    i = i + 1
    Loop
Cells(1, 6).Value = "D - máx"
For i = N To N 'imprime el D- mas grande
Cells(2, 6).Value = dmenos(i)
Next i
Cells(1, 7).Value = "valor de D máx"
```

```
If Cells(2, 5).Value > Cells(2, 6).Value Then 'imprime el D max

Cells(2, 7).Value = Cells(2, 5).Value

ElseIf Cells(2, 5).Value < Cells(2, 6).Value Then

Cells(2, 7).Value = Cells(2, 6).Value

End If

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

End Sub
```

4.1 Descripción del código fuente.

El presente código o algoritmo tiene como finalidad agrupar los números aleatorios de forma ascendente para posteriormente calcular el estadístico de Kolmogorov-Smirnov, y cuyo resultado se imprime en la hoja de cálculo (Excel).

A continuación se hace una lista de todas las variables que se utilizó en la elaboración del código fuente.

Variables	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
sorteado	Boolean	Condición
j	Integer	Contador
aux	Double	Almacena temporalmente los datos
dmas()	Double	Vector, almacena datos
dmenos()	Double	Vector, almacena datos

- ✓ Declaración de variables.
- ✓ Solicitud de valores (parámetros) necesarias durante la ejecución de programa.
- ✓ Generación de N números aleatorios.
- ✓ Ordenamiento de los N números aleatorios para ello se utiliza el método de la burbuja y los datos ordenados (números aleatorios) se imprimen en la hoja de cálculo(Excel), en la columna B.
- ✓ Se realiza un ciclo, como se muestra en el código fuente para calcular el D- y el D+ para cada número aleatorio ordenado.

- ✓ Se encuentra el D- máximo y el D+ máximo. Se hace un ciclo que permita ordenar de forma ascendente, los D- y los D+ y posteriormente se imprimen los últimos números ordenados, que vienen siendo el D- máximo y el D+ máximo.
- ✓ A partir de los dos números obtenidos (D- máximo y D+ máximo), se obtiene el D máximo de estos dos números o datos, y que viene siendo el estadístico de Kolmogorov-Smirnov.
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

4.2 Valor crítico de la distribución de Kolmogorov-Smirnov (α=5%).

D (N=5000, α=5%)=1.36

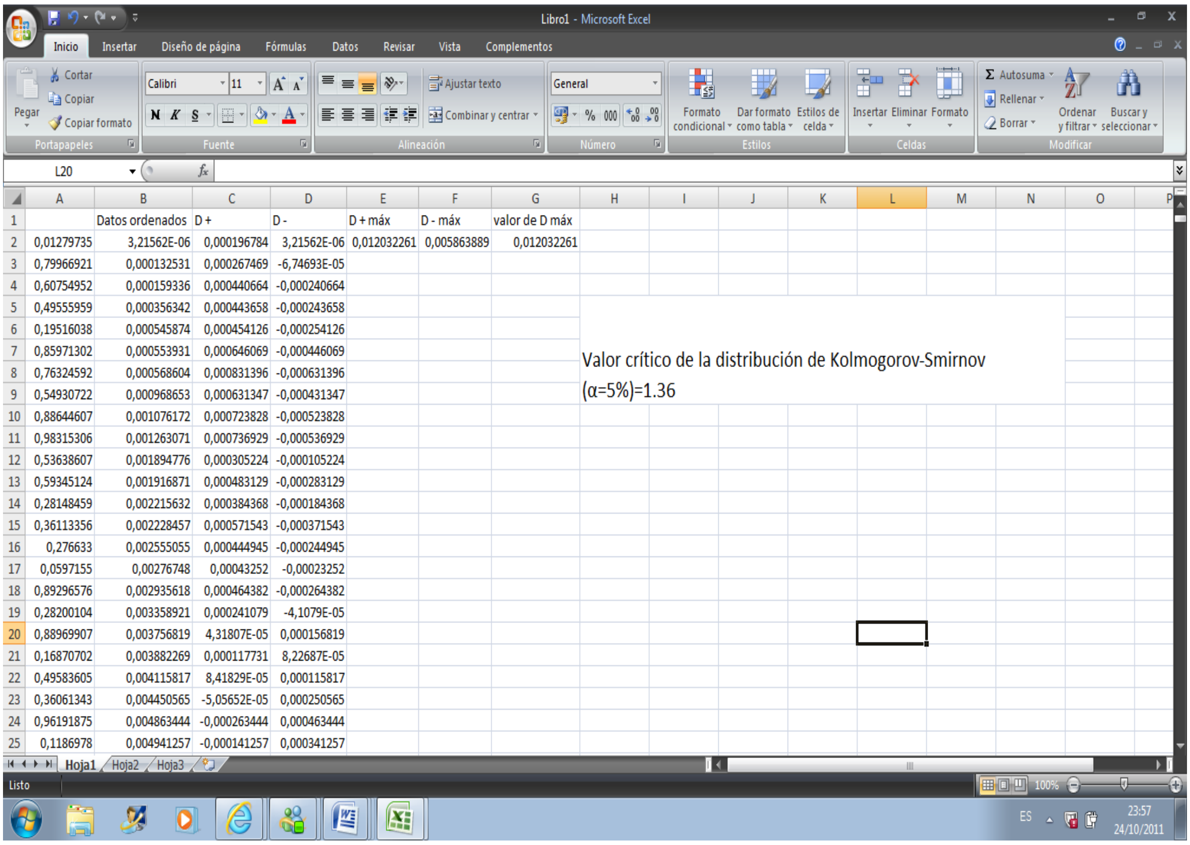
"este dato se encontró en la tabla de distribución de Kolmogorov-Smirnov"

5 Juego de parámetros.

Nota: en todos los casos el valor de la semilla x0=245, N=5,000 números aleatorios.

5.1 Caso 1.

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



	A	B	C	D	E	F	G
1		Datos ordenados	D +	D -	D + máx	D - máx	valor de D máx
2	0,01279735	3,21562E-06	0,000196784	3,21562E-06	0,012032261	0,005863889	0,012032261
3	0,79966921	0,000132531	0,000267469	-6,74693E-05			
4	0,60754952	0,000159336	0,000440664	-0,000240664			
5	0,49555959	0,000356342	0,000443658	-0,000243658			
6	0,19516038	0,000545874	0,000454126	-0,000254126			
7	0,85971302	0,000553931	0,000646069	-0,000446069			
8	0,76324592	0,000568604	0,000831396	-0,000631396			
9	0,54930722	0,000968653	0,000631347	-0,000431347			
10	0,88644607	0,001076172	0,000723828	-0,000523828			
11	0,98315306	0,001263071	0,000736929	-0,000536929			
12	0,53638607	0,001894776	0,000305224	-0,000105224			
13	0,59345124	0,001916871	0,000483129	-0,000283129			
14	0,28148459	0,002215632	0,000384368	-0,000184368			
15	0,36113356	0,002228457	0,000571543	-0,000371543			
16	0,276633	0,002555055	0,000444945	-0,000244945			
17	0,0597155	0,00276748	0,00043252	-0,00023252			
18	0,89296576	0,002935618	0,000464382	-0,000264382			
19	0,28200104	0,003358921	0,000241079	-4,1079E-05			
20	0,88969907	0,003756819	4,31807E-05	0,000156819			
21	0,16870702	0,003882269	0,000117731	8,22687E-05			
22	0,49583605	0,004115817	8,41829E-05	0,000115817			
23	0,36061343	0,004450565	-5,05652E-05	0,000250565			
24	0,96191875	0,004863444	-0,000263444	0,000463444			
25	0,1186978	0,004941257	-0,000141257	0,000341257			

Valor crítico de la distribución de Kolmogorov-Smirnov (α=5%)=1.36

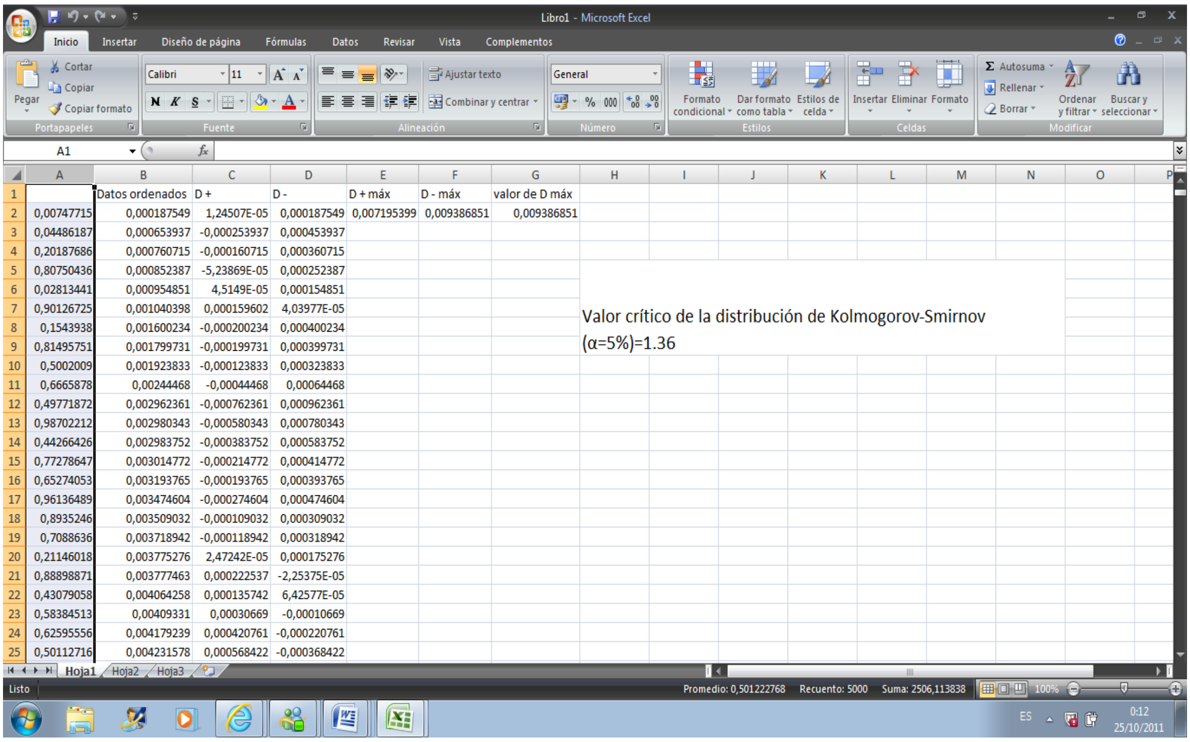
Figura 5: simulación realizada en VBA con los parámetros citados (Caso 1) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor del estadístico el de Kolmogorov-Smirnov es D= 0.012032261.

Entonces. D= 0.012032261.

0.012032261< 1.36 Por lo tanto: “No se rechaza la hipótesis nula”

5.2 Caso 2.

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



	A	B	C	D	E	F	G
1		Datos ordenados	D +	D -	D + máx	D - máx	valor de D máx
2	0,00747715	0,000187549	1,24507E-05	0,000187549	0,007195399	0,009386851	0,009386851
3	0,04486187	0,000653937	-0,000253937	0,000453937			
4	0,20187686	0,000760715	-0,000160715	0,000360715			
5	0,80750436	0,000852387	-5,23869E-05	0,000252387			
6	0,02813441	0,000954851	4,5149E-05	0,000154851			
7	0,90126725	0,001040398	0,000159602	4,03977E-05			
8	0,1543938	0,001600234	-0,000200234	0,000400234			
9	0,81495751	0,001799731	-0,000199731	0,000399731			
10	0,5002009	0,001923833	-0,000123833	0,000323833			
11	0,6665878	0,00244468	-0,00044468	0,00064468			
12	0,49771872	0,002962361	-0,000762361	0,000962361			
13	0,98702212	0,002980343	-0,000580343	0,000780343			
14	0,44266426	0,002983752	-0,000383752	0,000583752			
15	0,77278647	0,003014772	-0,000214772	0,000414772			
16	0,65274053	0,003193765	-0,000193765	0,000393765			
17	0,96136489	0,003474604	-0,000274604	0,000474604			
18	0,8935246	0,003509032	-0,000109032	0,000309032			
19	0,7088636	0,003718942	-0,000118942	0,000318942			
20	0,21146018	0,003775276	2,47242E-05	0,000175276			
21	0,88898871	0,003777463	0,000222537	-2,25375E-05			
22	0,43079058	0,004064258	0,000135742	6,42577E-05			
23	0,58384513	0,00409331	0,00030669	-0,00010669			
24	0,62595556	0,004179239	0,000420761	-0,000220761			
25	0,50112716	0,004231578	0,000568422	-0,000368422			

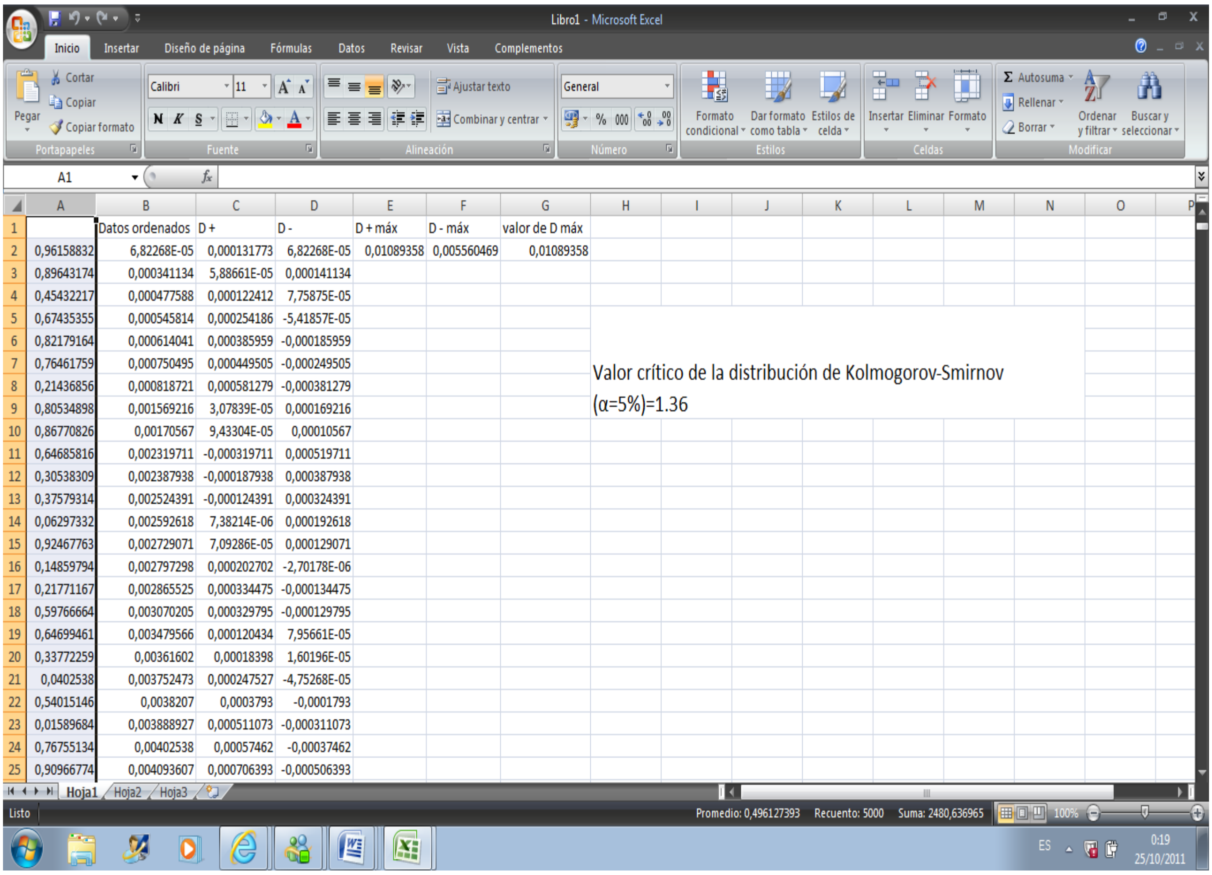
Figura 6: simulación realizada en VBA con los parámetros citados (Caso 2) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor del estadístico el de Kolmogorov-Smirnov es D= 0.009386851.

Entonces. D= 0.009386851

0.009386851< 1.36 Por lo tanto: “No se rechaza la hipótesis nula”

5.3 Caso 3.

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



	A	B	C	D	E	F	G
1		Datos ordenados	D +	D -	D + máx	D - máx	valor de D máx
2	0,96158832	6,82268E-05	0,000131773	6,82268E-05	0,01089358	0,005560469	0,01089358
3	0,89643174	0,000341134	5,88661E-05	0,000141134			
4	0,45432217	0,000477588	0,000122412	7,75875E-05			
5	0,67435355	0,000545814	0,000254186	-5,41857E-05			
6	0,82179164	0,000614041	0,000385959	-0,000185959			
7	0,76461759	0,000750495	0,000449505	-0,000249505			
8	0,21436856	0,000818721	0,000581279	-0,000381279			
9	0,80534898	0,001569216	3,07839E-05	0,000169216			
10	0,86770826	0,00170567	9,43304E-05	0,00010567			
11	0,64685816	0,002319711	-0,000319711	0,000519711			
12	0,30538309	0,002387938	-0,000187938	0,000387938			
13	0,37579314	0,002524391	-0,000124391	0,000324391			
14	0,06297332	0,002592618	7,38214E-06	0,000192618			
15	0,92467763	0,002729071	7,09286E-05	0,000129071			
16	0,14859794	0,002797298	0,000202702	-2,70178E-06			
17	0,21771167	0,002865525	0,000334475	-0,000134475			
18	0,59766664	0,003070205	0,000329795	-0,000129795			
19	0,64699461	0,003479566	0,000120434	7,95661E-05			
20	0,33772259	0,00361602	0,00018398	1,60196E-05			
21	0,0402538	0,003752473	0,000247527	-4,75268E-05			
22	0,54015146	0,0038207	0,0003793	-0,0001793			
23	0,01589684	0,003888927	0,000511073	-0,000311073			
24	0,76755134	0,00402538	0,00057462	-0,00037462			
25	0,90966774	0,004093607	0,000706393	-0,000506393			

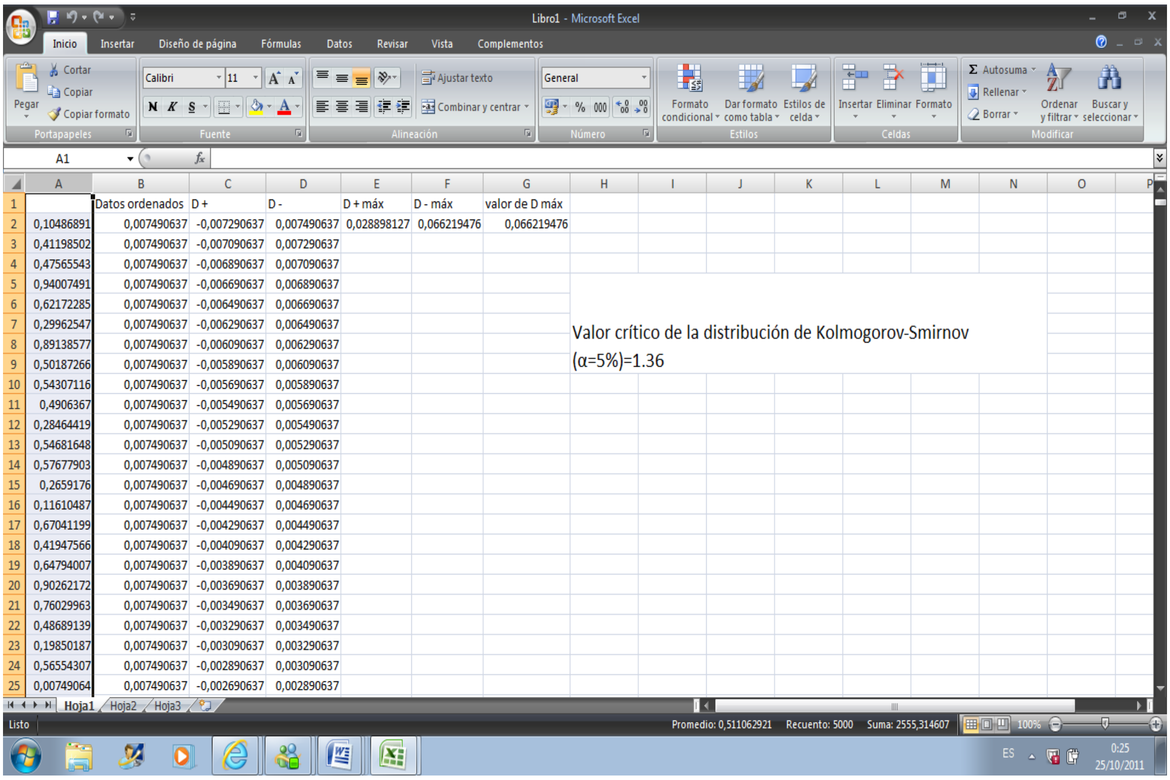
Figura 7: simulación realizada en VBA con los parámetros citados (Caso 3) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor del estadístico el de Kolmogorov-Smirnov es D= 0.01089358.

Entonces. D= 0.01089358

0,01089358 < 1.36 Por lo tanto: "No se rechaza la hipótesis nula"

5.4 Caso 4

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



	A	B	C	D	E	F	G
1		Datos ordenados	D +	D -	D + máx	D - máx	valor de D máx
2	0,10486891	0,007490637	-0,007290637	0,007490637	0,028898127	0,066219476	0,066219476
3	0,41198502	0,007490637	-0,007090637	0,007290637			
4	0,47565543	0,007490637	-0,006890637	0,007090637			
5	0,94007491	0,007490637	-0,006690637	0,006890637			
6	0,62172285	0,007490637	-0,006490637	0,006690637			
7	0,29962547	0,007490637	-0,006290637	0,006490637			
8	0,89138577	0,007490637	-0,006090637	0,006290637			
9	0,50187266	0,007490637	-0,005890637	0,006090637			
10	0,54307116	0,007490637	-0,005690637	0,005890637			
11	0,4906367	0,007490637	-0,005490637	0,005690637			
12	0,28464419	0,007490637	-0,005290637	0,005490637			
13	0,54681648	0,007490637	-0,005090637	0,005290637			
14	0,57677903	0,007490637	-0,004890637	0,005090637			
15	0,2659176	0,007490637	-0,004690637	0,004890637			
16	0,11610487	0,007490637	-0,004490637	0,004690637			
17	0,67041199	0,007490637	-0,004290637	0,004490637			
18	0,41947566	0,007490637	-0,004090637	0,004290637			
19	0,64794007	0,007490637	-0,003890637	0,004090637			
20	0,90262172	0,007490637	-0,003690637	0,003890637			
21	0,76029963	0,007490637	-0,003490637	0,003690637			
22	0,48689139	0,007490637	-0,003290637	0,003490637			
23	0,19850187	0,007490637	-0,003090637	0,003290637			
24	0,56554307	0,007490637	-0,002890637	0,003090637			
25	0,00749064	0,007490637	-0,002690637	0,002890637			

Valor crítico de la distribución de Kolmogorov-Smirnov
(α=5%)=1.36

Figura 8: simulación realizada en VBA con los parámetros citados (Caso 4) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El valor del estadístico el de Kolmogorov-Smirnov es D= 0.066219476.

Entonces. D= 0.066219476

0.066219476 < 1.36 Por lo tanto: "No se rechaza la hipótesis nula"

6 Conclusión sobre el rechazo o no de la hipótesis de uniformidad.

En todos los casos se acepta la hipótesis nula, debido que no existen diferencias significativas entre la distribución teórica (con un nivel de significación de 5%) y la distribución simulada por los resultados muéstrales, es decir, que los valores generados siguen la distribución que se había supuesto (uniforme).

<u>Pruebas de rachas</u>

7 *Rachas crecientes/decrecientes*

7.1 Código fuente.

```
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, aux As Double

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

Dim y() As Double, rachas As Integer

'pedir valores al usuario de cada uno de las variables

N = InputBox("numero de pseudoaleatorios 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 apartir de 1 hasta N numeros de pseudoaleatorios a generar

For i = 1 To N

x = Int((a * x0 + c) / m) 'realizar la siguiente operación

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

u(i) = x1 / m

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

Cells(i + 1, 1).Value = u(i) 'se imprime todos los resultados la columna A1

Next i

ReDim y(N) As Double ' se covierte de u(i) a y(i)

Cells(1, 3).Value = "Y(i)"

For i = 1 To N - 1

If u(i) < u(i + 1) Then 'y(i)=1  si u(i) es menor a u(i+1)

y(i) = 1

Cells(i + 1, 3).Value = y(i)
```

```
ElseIf u(i) > u(i + 1) Then 'y(i)=0 en caso contrario

y(i) = 0

Cells(i + 1, 3).Value = y(i)

End If

Next i

Cells(1, 4).Value = "Rachas"

rachas = 0 'conteo de rachas

For i = 1 To N - 2

If y(i) <> y(i + 1) Then

rachas = rachas + 1

End If

Next i

Cells(2, 4).Value = rachas

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

End Sub
```

7.2 Descripción del código fuente.

La función principal de este programa es contar el número de rachas (crecientes/decrecientes) que existen en N números aleatorios generados y a partir de esto observar si existe o no la independencia de los mismos.

A continuación se hace una lista de todas las variables que se utilizó en la elaboración del código fuente.

Variables	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
y()	Doublé	Vector, almacena datos

rachas	Integer	contador

- ✓ Declaración de variables.
- ✓ Solicitud de valores (parámetros) necesarias durante la ejecución de programa.
- ✓ Generación de N números aleatorios u(i).
- ✓ Transformación de u(i) (N números aleatorios) a Y(i) (N-1 números aleatorios), y(i)=1 cuando u(i) es menor a u (i+1) y Y(i)=0 en caso contrario.
- ✓ Conteo de rachas. Se va aumentando el número de rachas cuando Y (i) es diferente de Y (i+1).
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

7.3 Detalles de la prueba de hipótesis.

El número de rachas debe tener la siguiente distribución.

N $\left(\mu = \frac{2N-1}{3}, \sigma^2 = \frac{16N-1}{90}\right)$ donde:

μ=media muestral ; σ^2=varianza

A partir de N=5000 números aleatorios tenemos:

NR ~N (μ=3,333, σ=29.814) NR=numero de rachas.

Con α=5%, tenemos:

P(a<=NR<=b)=95%

-1.96<=NR<=1.96

3,333 -1.96 (29.814) <=NR<=3,333 +1.96 (29.814)

3,274.56<=NR<=3,391.43 “Intervalo de confianza”

8 Juego de parámetros.

Nota: en todos los casos el valor de la semilla x0=245, N=5,000 números aleatorios.

8.1 Caso 1

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

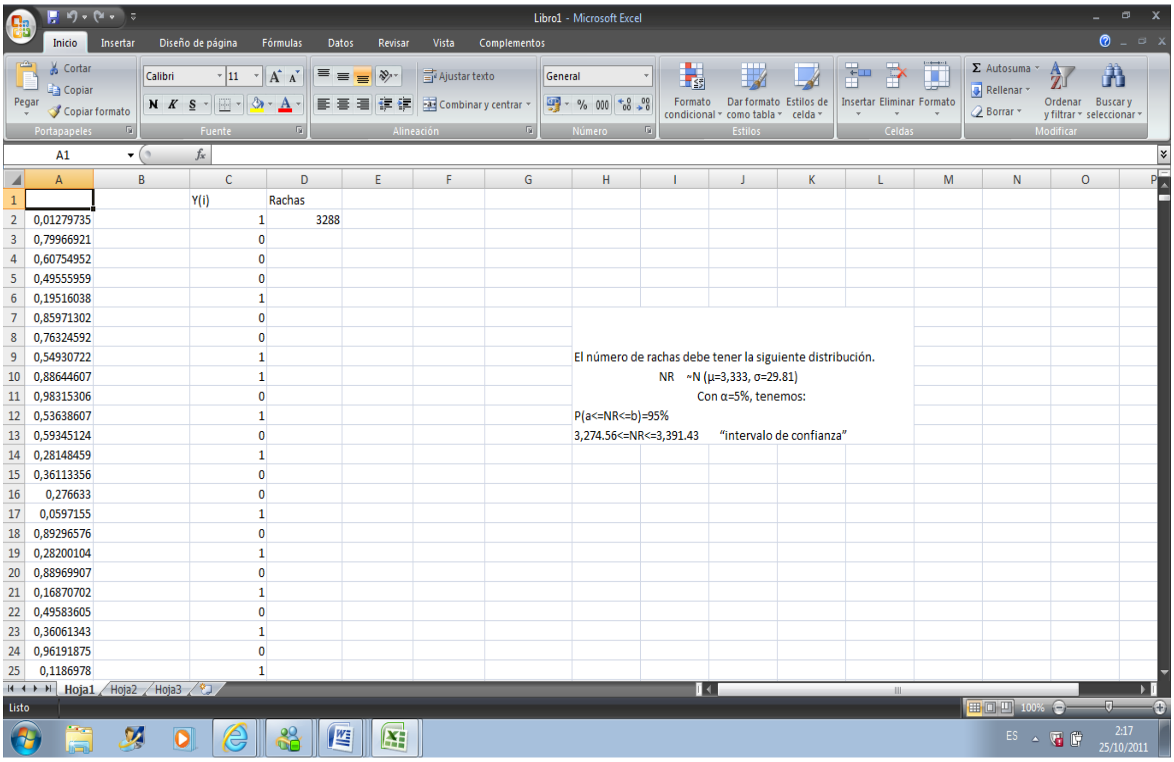


Figura 9: simulación realizada en VBA con los parámetros citados (Caso 1) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 3288

Se acepta la hipótesis de independencia con un nivel de significación α=5%.

8.2 Caso 2

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

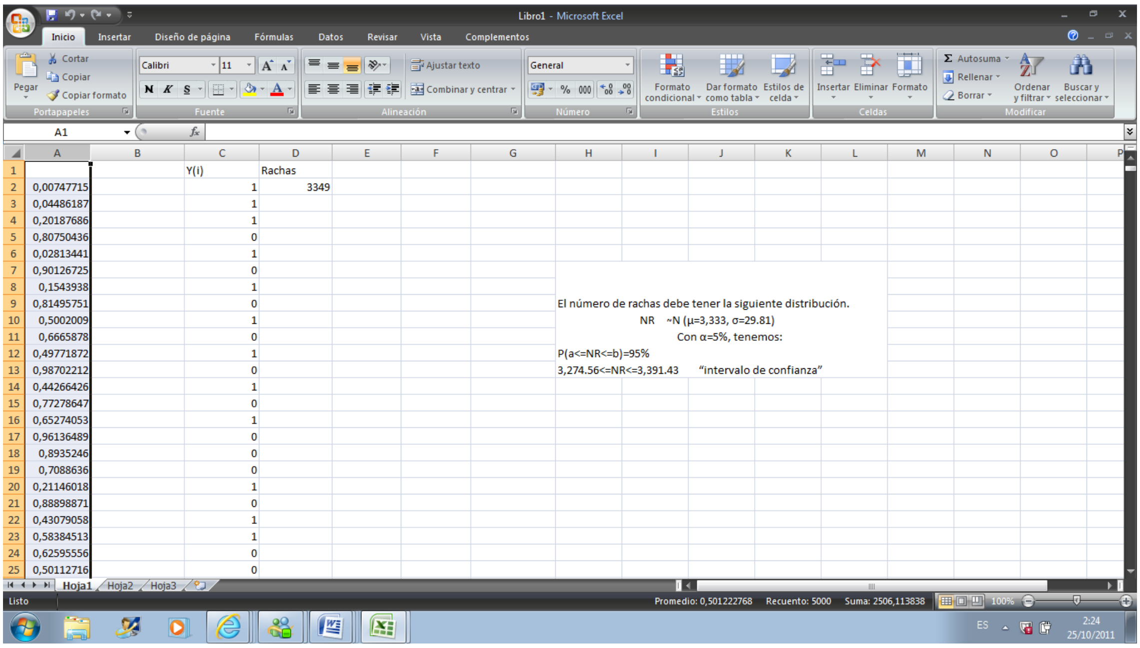


Figura 10: simulación realizada en VBA con los parámetros citados (Caso 2) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 3349 “se acepta la hipótesis”

8.3 Caso 3.

a=237, c=0 y m=14657

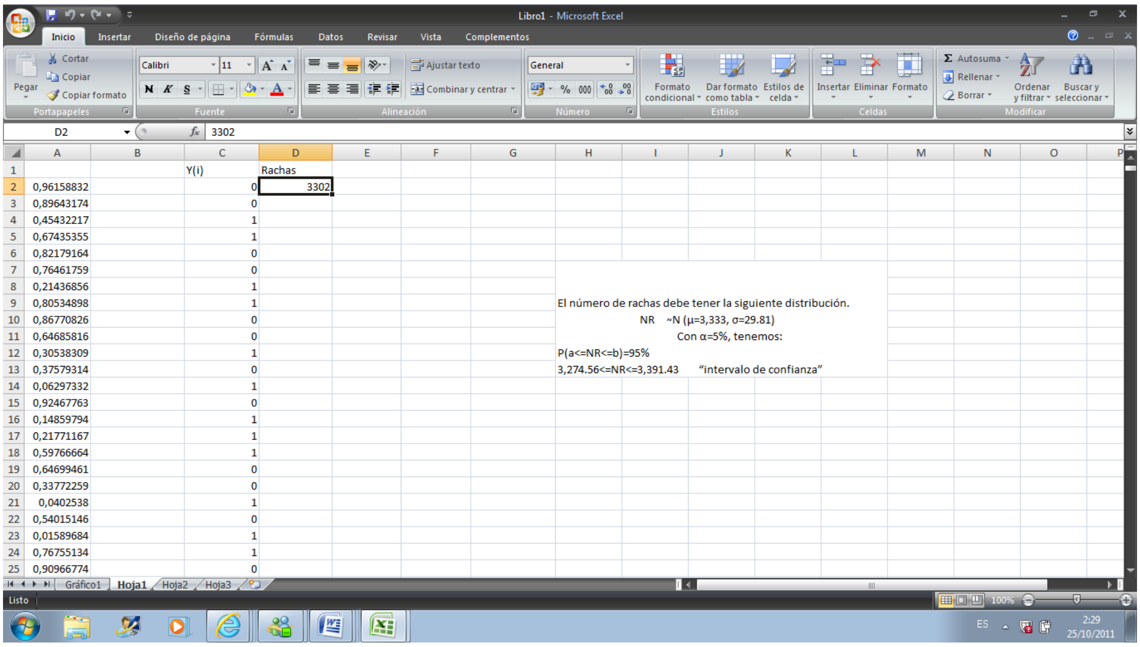


Figura 11: simulación realizada en VBA con los parámetros citados (Caso 3) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 3302 “se acepta la hipótesis”

8.4 Caso 4

a=23, c=0 y m=267

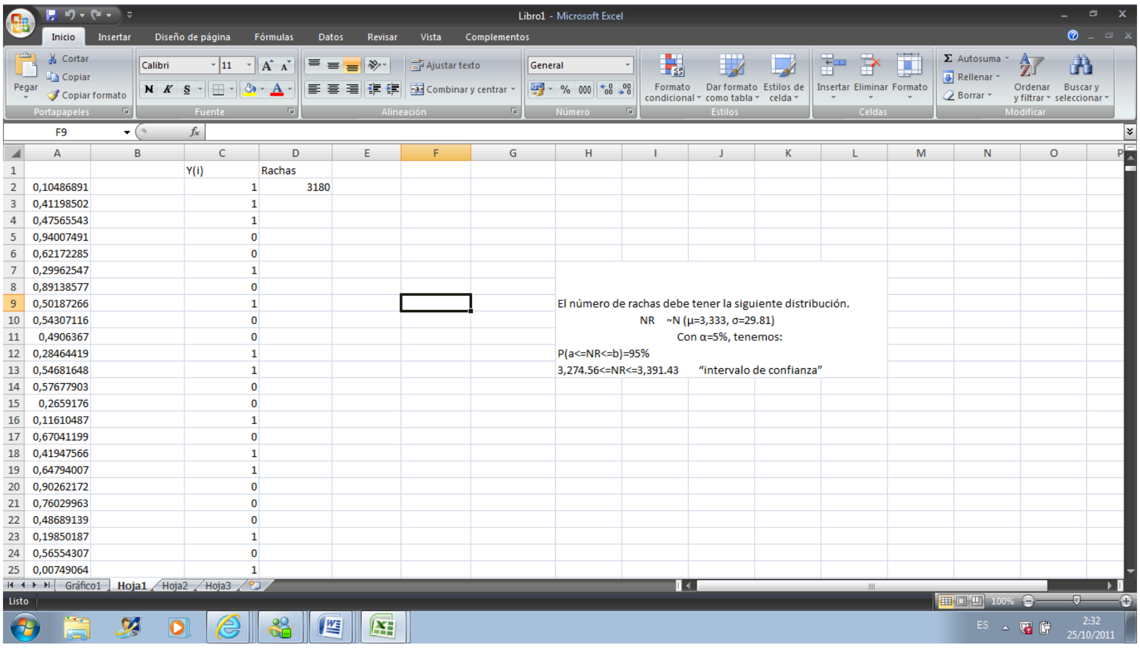


Figura 12: simulación realizada en VBA con los parámetros citados (Caso 4) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 3180 "se rechaza la hipótesis"

9 Conclusión sobre el rechazo o no de la hipótesis de independencia.

Los casos 1, 2 y 3 se acepta la hipótesis de independencia por el numero de rachas se encuentra en el intervalo de confianza con un nivel de significación de 5%, mientras que el caso 4 se rechaza o no se acepta la hipótesis de independencia por que el numero de rachas (NR) no se encuentra en el intervalo de confianza.

10 Rachas por encima/debajo de la media.

10.1 Código fuente.

```
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, n1 As Integer, n2 As Integer

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

Dim y() As Double, rachas As Integer

'pedir valores al usuario de cada uno de las variables

N = InputBox("numero de pseudoaleatorios 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 apartir de 1 hasta N numeros de pseudoaleatorios a generar

For i = 1 To N

x = Int((a * x0 + c) / m) 'realizar la siguiente operación

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

u(i) = x1 / m

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

Cells(i + 1, 1).Value = u(i) 'se imprime todos los resultados la columna A1

Next i
```

```
'ReDim aux(N) As Double

ReDim y(N) As Double

n1 = 0

n2 = 0

Cells(1, 3) = "Y(i)"

For i = 1 To N  'ciclo que permite pasar de u(i) a y(i) y ademas el numero de n1 y n2

If u(i) < 0.5 Then   'si u(i) es menor de

y(i) = 1

n1 = n1 + 1

Cells(i + 1, 3).Value = y(i)

ElseIf u(i) > 0.5 Then

y(i) = 0

n2 = n2 + 1

Cells(i + 1, 3).Value = y(i)

End If

Next i

Cells(1, 5).Value = "n1"

Cells(1, 6).Value = "n2"

Cells(2, 5).Value = n1

Cells(2, 6).Value = n2

rachas = 0

For i = 1 To N - 1 'ciclo que permite contar en numero de rachas que existe

If y(i) <> y(i + 1) Then

rachas = rachas + 1

End If

Next i

Cells(1, 4).Value = "Rachas"

Cells(2, 4).Value = rachas

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

End Sub
```

10.2 Descripción del código fuente.

La finalidad de este programa es contar el número de rachas (*por encima/debajo de la media*) que existen en N números aleatorios generados y a partir de esto observar si existe o no la independencia de los mismos.

A continuación se hace una lista de todas las variables que se utilizó en la elaboración del código fuente.

Variables	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
y()	Doublé	Vector, almacena datos
rachas	Integer	contador
n1	Integer	Contador
n2	Integer	Contador

- ✓ Declaración de variables.
- ✓ Solicitud de valores (parámetros) necesarias durante la ejecución de programa.
- ✓ Generación de N números aleatorios u(i).
- ✓ Transformación de u(i) (N números aleatorios) a Y(i) (N números aleatorios), y(i)=1 cuando u(i) es menor a 0.5 y Y(i)=0 en caso contrario. En este mismo ciclo se cuentan el numero de datos debajo de la media (n1) y se va aumentando cuando u(i) es menor a 0.5. También se cuentan el numero de datos arriba de la media y se aumenta cada vez que u(i) sea mayor a 0.5.
- ✓ Conteo de rachas. Se va aumentando el número de rachas cuando Y (i) es diferente de Y (i+1).
- ✓ Y por último aparece una ventana de Windows que indica que la ejecución del programa ha terminado exitosamente.

10.3 Detalles de la prueba de hipótesis.

El número de rachas debe tener la siguiente distribución.

$$N\left(\mu = \frac{2n1n2}{N} + \frac{1}{2}, \sigma^2 = \frac{2n1n2(2n1n2-N)}{N^2(N-1)}\right)$$

donde:

μ=media muestral ; σ^2=varianza; n1=numero de datos debajo de la media; n2=numero de datos arriba de la media.

11 Juego de parámetros

Nota: en todos los casos el valor de la semilla x0=245, N=5,000 números aleatorios.

11.1 Caso 1

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

A partir de N=5000 números aleatorios tenemos, n1= 2530, n2= 2470:

NR ~N (μ=2,500.14, σ=1249.38) NR=numero de rachas.

Con α=5%, tenemos:

P(a<=NR<=b)=95%

-1.96<=NR<=1.96

51.3552<=NR<=4948.9248 “Intervalo de confianza”

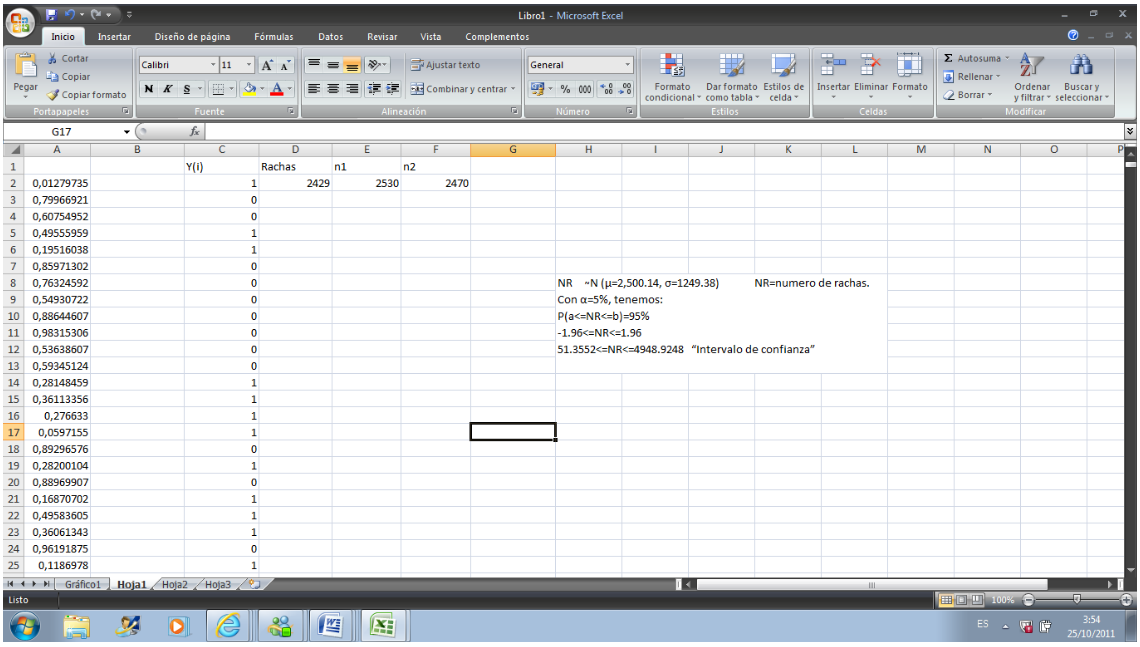


Figura 13: simulación realizada en VBA con los parámetros citados (Caso 1) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 2429 “se acepta la hipótesis”

11.2 Caso 2.

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

A partir de N=5000 números aleatorios tenemos, n1= 2479, n2= 2521:

NR ~N (μ=2,500.32, σ=1249.57) NR=numero de rachas.

Con α=5%, tenemos:

P(a<=NR<=b)=95%

-1.96<=NR<=1.96

51.1628<=NR<=4949.4772 “Intervalo de confianza”

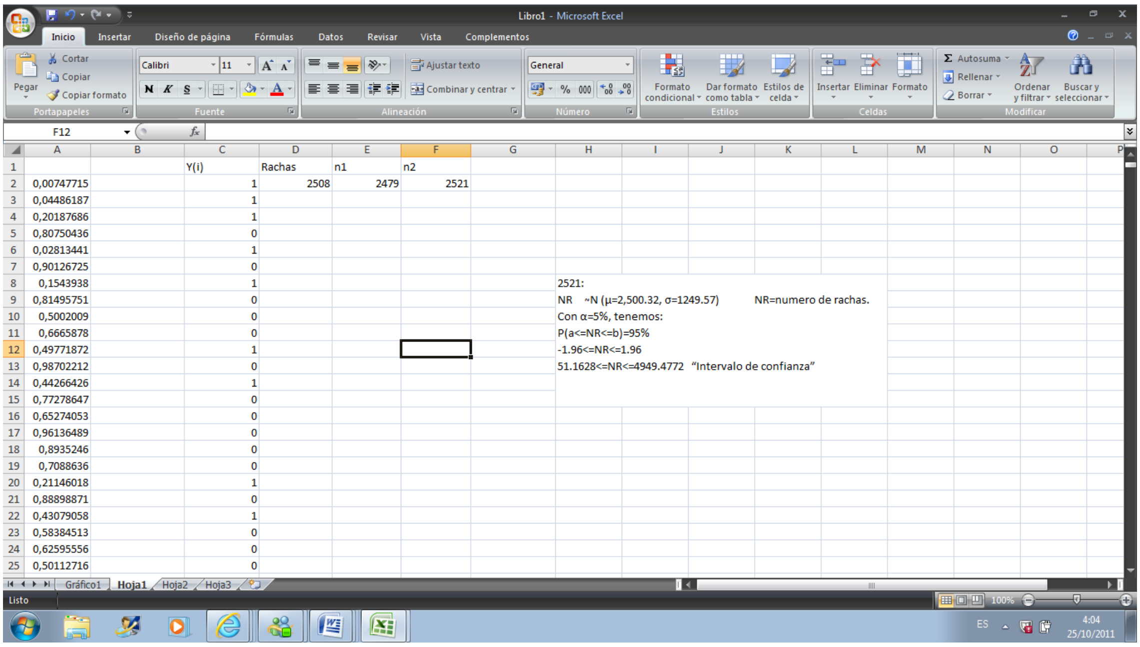


Figura 14: simulación realizada en VBA con los parámetros citados (Caso 2) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 2508 “se acepta la hipótesis”

11.3 Caso 3.

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

A partir de N=5000 números aleatorios tenemos, n1= 2547, n2= 2453:

NR ~N (μ=2,499.61, σ=1248.86) NR=numero de rachas.

Con α=5%, tenemos:

P(a<=NR<=b)=95%

-1.96<=NR<=1.96

51.84<=NR<=4947.37 “Intervalo de confianza”

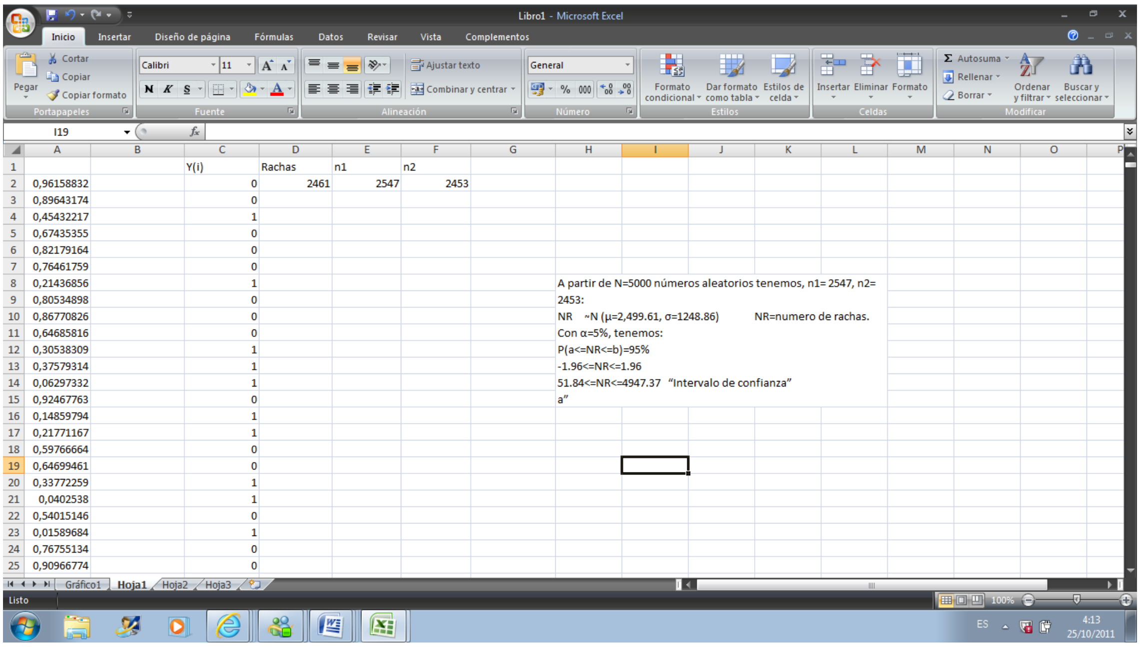


Figura 15: simulación realizada en VBA con los parámetros citados (Caso 3) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 2461 "se acepta la hipótesis"

11.4 Caso 4

a=23, c=0 y m=267

A partir de N=5000 números aleatorios tenemos, n1= 2333, n2= 2667:

NR ~N (µ=2,489.34, σ=1238.62) NR=numero de rachas.

Con α=5%, tenemos:

P(a<=NR<=b)=95%

-1.96<=NR<=1.96

61.64<=NR<=4917.03 "Intervalo de confianza"

Figura 15: simulación realizada en VBA con los parámetros citados (Caso 3) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 2461 "se acepta la hipótesis"

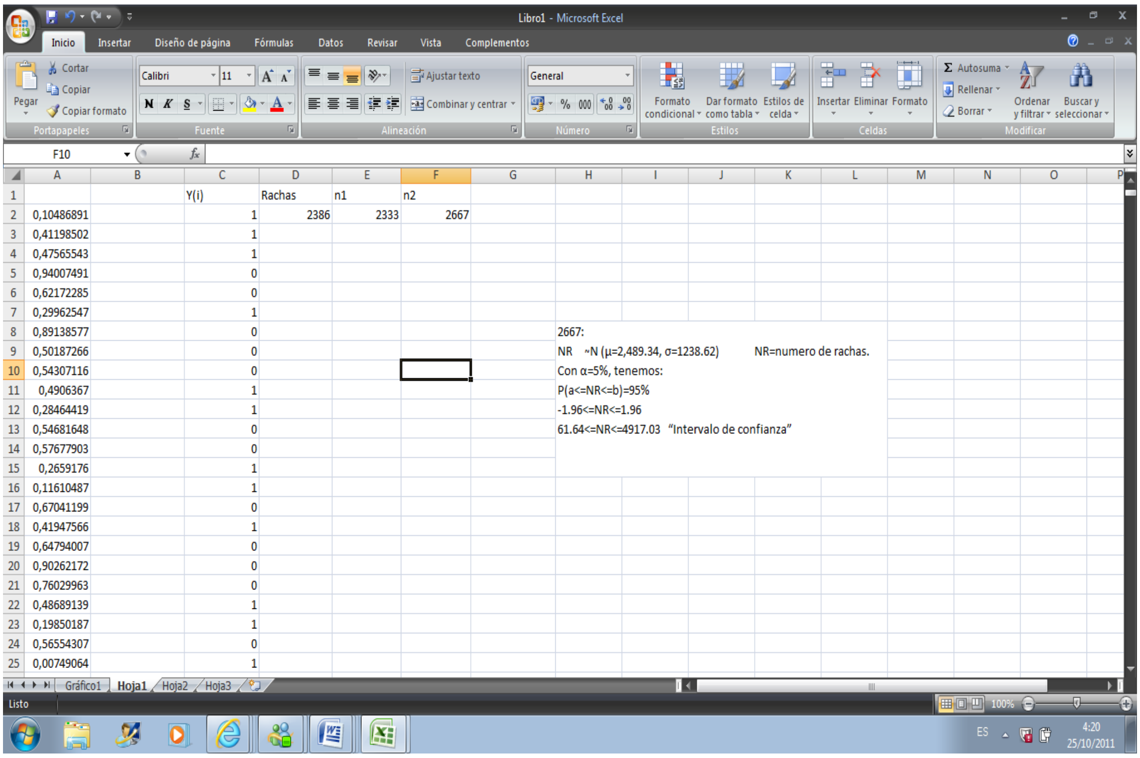


Figura 16: simulación realizada en VBA con los parámetros citados (Caso 4) y los números aleatorios se imprimen en la hoja de cálculo (Excel). El numero de rachas es NR= 2386 "se acepta la hipótesis"

12. Conclusión sobre el rechazo o no de la hipótesis de independencia.

En todos los casos se acepta la hipótesis de independencia, esto es porque el número de rachas siempre se encuentra en el intervalo de confianza con un nivel de significación de 5%. Y con esto decimos que los números aleatorios que se están generando en todos los casos son independientes.