

PROPUESTA PARA GENERAR NÚMEROS ALEATORIOS EN HOJAS DE CÁLCULO

PROPOSAL TO GENERATE RANDOM NUMBERS IN SPREADSHEETS

Celeste Escutia Alvarado

Tecnológico Nacional de México / IT de Celaya, México
M2103002@itcelaya.edu.mx

Manuel Darío Hernández Ripalda

Tecnológico Nacional de México / IT de Celaya, México
darío.hernandez@itcelaya.edu.mx

José Alfredo Jiménez García

Tecnológico Nacional de México / IT de Celaya, México
alfredo.jimenez@itcelaya.edu.mx

Recepción: 28/mayo/2022

Aceptación: 22/noviembre/2022

Resumen

Las ciencias computacionales y los lenguajes de programación están cimentados sobre las leyes y reglas matemáticas. Un ejemplo son los números aleatorios. Las aplicaciones de simulación dependen de la calidad y confiabilidad de estos números.

Se elaboró una aplicación para usar en cualquier hoja de cálculo con formato xls, con el fin de que dicha aplicación sea compatible y ejecutable en cualquier sistema operativo sin restricciones de hardware y software, y a su vez, pueda generar secuencias replicables de números aleatorios.

Luego de buscar entre varios métodos, de generación de aleatorios. Hay uno que usa C++, que es una variación del Lehmer; posteriormente se evaluó, con pruebas estándares como la autocorrelación.

Los números aleatorios tienen importancia hoy en día para diferentes procesos, tanto académicos como de campo, especialmente aquellos que necesitan cantidades importantes en el sistema, y son una clave en su funcionamiento.

Palabras Clave: Generador de números aleatorios, hojas de cálculo, pruebas de aleatoriedad, reproducibilidad.

Abstract

Computer science and programming languages are founded on mathematical laws and rules. An example is random numbers. Simulation applications depend on the quality and reliability of these numbers.

An application was developed to be used in any spreadsheet with xls format, so that said application is compatible and executable in any operating system without hardware and software restrictions, and in turn, can generate replicable sequences of numbers. random.

After searching among several methods, of random generation. There is one that uses C++, which is a variation of Lehmer; then it was evaluated, with standard tests such as autocorrelation.

Random numbers are important today for different processes, both academic and field, especially those that need large quantities in the system, and are a key to their operation.

Keywords: *Random number generator, spreadsheets, randomness tests, reproducibility.*

1. Introducción

Los objetivos de esta investigación es identificar un método de generación de números pseudoaleatorios, para uso en simulación, que cumpla con:

- Cualquier persona, empleando los pasos descritos, puede llevarlo a cabo en cualquier hoja de cálculo, como en un Google sheets, hoja de cálculo de Excel, hoja "calc" de LibreOffice o de Mac. Lo que implica que se pueda correr en diferentes sistemas operativos.
- Que las secuencias de números generados sean reproducibles. El comando actual en las hojas de cálculo (aleatorio ()), es una función volátil y está cambiando las secuencias.
- El método debe ser fácil de implementar y editar.
- Debe ser igual en desempeño estadístico básico al generador disponible actualmente en las hojas de cálculo.

Se ha discutido el concepto y la existencia de la verdadera aleatoriedad durante muchos años. La mecánica cuántica nos ha demostrado que existe en el mundo real y son una parte esencial de las reglas que gobiernan nuestro universo [Dennis & Johnson, 2007].

La generación de números pseudoaleatorios utiliza algoritmos de fácil manejo en computadoras digitales, las secuencias de números deben cumplir razonablemente las pruebas de aleatoriedad.

En la simulación Monte Carlo es preferible contar con una secuencia de números pseudoaleatorios que permita la depuración y se vea facilitada por la capacidad de ejecutar la misma secuencia de números aleatorios nuevamente, comenzando desde la misma semilla aleatoria. Esta capacidad se llama reproducibilidad. También se utilizan en criptografía, siempre que la semilla de generación sea secreta. El remitente y el receptor pueden generar el mismo conjunto de números automáticamente para usarlos como claves.

En la tabla 1 se presenta un resumen en forma de los varios procedimientos para generar números pseudoaleatorios:

- Método de medios cuadrados: En matemáticas, el método del cuadrado medio es un método para generar números pseudoaleatorios. En la práctica no es un buen método, ya que su período suele ser muy corto y tiene algunas debilidades severas; repetido suficientes veces, el método del cuadrado central comenzará a generar repetidamente el mismo número o pasará a un número anterior en la secuencia y se repetirá indefinidamente [Von Neumann, 1951].

El método fue inventado por John Von Neumann y fue descrito en una conferencia en 1949. Mencionó que no había verdaderos "números aleatorios", sólo medios para producirlos, y "un procedimiento aritmético estricto", como el método del cuadrado medio, "no es tal método". Sin embargo, descubrió que estos métodos eran cientos de veces más rápidos que leer números "verdaderamente".

- Generador de Lehmer: También denominado generador de números aleatorios de Park y Miller. La serie de valores generados por dichos algoritmos

generalmente está determinada por un número fijo llamado semilla. Es un tipo de generador lineal congruente que opera en grupo multiplicativo de enteros módulo n . La fórmula general se puede observar en ecuación 1.

$$X_{K+1} = a * X_{K+1} \bmod m \quad (1)$$

Donde a es el número fijo llamado semilla, el módulo m es una operación que calcula el residuo de la división entre una cantidad y el número m , por ejemplo, 22 módulo 7, resulta en 1 [Lehmer, 1951].

Tabla 1 Generadores de números pseudoaleatorios.

Generador	Primeros proponentes y año	Nota
Método de medios cuadrados	J. Von Neumann, 1946	En su forma original, es de mala calidad y solo de interés histórico.
Generador de Lehmer	D.H. Lehmer, 1951	Uno de los diseños más antiguos e influyentes.
Generador lineal congruencial	W. E. Thomson; A. Rotenberg, 1958	Una generalización del generador de Lehmer e históricamente el generador más influyente y estudiado.
Generador Wichmann- Hill	B. A. Wichmann and D. I. Hill, 1982	Una combinación de tres LCG pequeños, adecuados para CPU de 16 bits. Ampliamente utilizado en muchos programas, p. se usa en Excel 2003 y versiones posteriores para la función de Excel RAND y fue el generador por defecto en el lenguaje Python hasta la versión 2.2.
Generador Park y Miller	S. K. Park and K. W. Miller, 1988	Una implementación específica de un generador Lehmer, muy utilizada porque está incluida en C++ como la función <code>minstd_rand0</code> desde C++11 en adelante.
Xorshift	G. Marsaglia, 2003	Es un subtipo muy rápido de generadores LFSR. Marsaglia también sugirió como mejora el generador xorwow, en el que se añade la salida de un generador xorshift con una secuencia de Weyl.

Fuente: *Elaboración propia.*

- **Wichmann- Hill:** Es un generador de números pseudoaleatorios propuesto en 1982 por Brian Wichmann y David Hill. Consiste en tres generadores lineales congruentes con diferentes módulos primos, cada uno de los cuales se usa para producir un número distribuido uniformemente entre 0 y 1. Estos se suman, módulo 1, para producir el resultado [Wichmann & Hill, 1982].

La suma de tres generadores produce una secuencia pseudoaleatoria con un ciclo superior a 6.95×10^{12} . Específicamente, los módulos son 30269, 30307 y 30323, produciendo períodos de 30268, 30306 y 30322. El período total es el mínimo común múltiplo de estos: $(30268 \times 30306 \times 30322) / 4 = 6,953,607,871,644$.

- Generador Park y Miller: En 1988, Park y Miller, sugirieron un Lehmer RNG con parámetros particulares $m = 2^{31} - 1 = 2,147,483,647$ (un primo de Mersenne M31) y $a = 7^5 = 16,807$ (un módulo raíz primitivo M31), ahora conocido como MINSTD. Aunque MINSTD fue posteriormente criticado por Marsaglia y Sullivan [1993], todavía se usa en la actualidad (en particular, en CarbonLib y minstd_rand0 de C++11). Park, Miller y Stockmeyer respondieron a las críticas, diciendo [Knuth, 1998]: "Dada la naturaleza dinámica del área, es difícil para los no especialistas tomar decisiones sobre qué generador usar. "Deme algo que pueda comprender, implementar y portar... no es necesario que sea lo último en tecnología, solo asegúrese de que sea razonablemente bueno y eficiente". Nuestro artículo y el generador estándar mínimo asociado fue un intento de responder a esta solicitud. Cinco años después, no vemos la necesidad de modificar nuestra respuesta más que sugerir el uso del multiplicador $a = 48271$ en lugar de 16807" [Park & Miller, 1988].
- Generación de números aleatorios a través de contadores: También conocida como generador de números pseudoaleatorios basado en contadores, es un tipo de generador de números pseudoaleatorios que utiliza solo un contador de enteros como su estado interno [Widynski, 2020].
Es posible pensar en un generador de números pseudoaleatorios (PRNG) como una función que transforma una serie de bits conocidos como el estado en un nuevo estado y un número aleatorio. Es decir, dada una función PRNG y un estado inicial estado (state 0) podemos usar repetidamente el PRNG para generar una secuencia de estados y números aleatorios. Las primeras pruebas para números aleatorios fueron publicadas por M.G. Kendall y Bernard Babington Smith en el Journal of the Royal Statistical Society en 1938. Estas se basaron en herramientas estadísticas como la prueba de Ji-cuadrado de Pearson que se desarrollaron para distinguir si los fenómenos experimentales coincidían con sus probabilidades teóricas. Pearson desarrolló su prueba originalmente, mostrando que una serie de experimentos con datos realizados por W.F.R. Weldon no mostró un comportamiento "aleatorio" [Kendall & Babington, 1938].

Las cuatro pruebas originales de Kendall y Smith eran pruebas de hipótesis, que tomaban como hipótesis nula la idea de que cada número en una secuencia aleatoria dada tenía la misma posibilidad de ocurrir, y que varios otros patrones en los datos también deberían distribuirse de manera equiprobable.

La prueba de frecuencia fue muy básica: verificar que hubiera aproximadamente la misma cantidad de 0, 1, 2, 3, etc.

La prueba serial, hizo lo mismo, pero para secuencias de dos dígitos a la vez (00, 01, 02, etc.), comparando sus frecuencias observadas con sus predicciones hipotéticas si estuvieran igualmente distribuidas.

La prueba de póquer, prueba ciertas secuencias de cinco números a la vez (AAAAA, AAAAB, AAABB, etc.) basada en las manos en el juego de póquer.

La prueba de la brecha analizó las distancias entre ceros (00 sería una distancia de 0, 030 sería una distancia de 1, 02250 sería una distancia de 3, etc.).

Métodos para probar secuencia de números pseudoaleatorios

Dado que un algoritmo genera una secuencia de números pseudoaleatorios surge la pregunta de qué tan cerca están ellos de ser aleatorios en su comportamiento. Para responder a este interrogante un gran número de pruebas estadísticas han sido propuestas:

- Prueba Ji-cuadrada. Esta prueba se interesa por el grado de acuerdo con que existen entre la distribución de una muestra de números aleatorios generados y la distribución uniforme. Está basada en la hipótesis nula.

H_0 : Los U_i son variables aleatorias independientes e idénticamente distribuidas con función de la distribución $U(0,1)$ [Astaiza, 1983].

La prueba Ji- Cuadrada en lugar de medir la diferencia de cada punto entre la muestra y la desviación verdadera, checa la desviación del valor esperado, como se ve en ecuación 2.

$$\sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

Donde:

n : Número de intervalos de clase

O_i : Número observado en la clase i

E_i : Número esperado en cada clase i

Para una distribución uniforme, E_i , el número en cada clase está dado por la ecuación 3.

$$E_i = \frac{N}{n} \quad (3)$$

Para clases igualmente espaciadas, donde N es el número total de observaciones. Puede ser mostrado que la distribución de la muestra se aproxima a la distribución Ji-Cuadrada con $n-1$ grados de libertad [Coss Bú, 2003].

- Prueba de correlación: Correlación es la relación recíproca entre dos o más cosas (elementos). A veces un grupo de números generados pueden parecer aleatorios, pero existe una relación entre cada cierto número de ellos a partir de alguno específico.
- Amplitud de autocorrelación: Es la distancia que existe entre los números de la lista que tiene la relación entre sí. Se da cada n -ésimo número aleatorio e inicia en el elemento i [Ríos & Ríos, 2000].
- Prueba de corridas: Una prueba de corridas es un método que nos ayuda a evaluar el carácter de aleatoriedad de una secuencia de números estadísticamente independientes y números uniformemente distribuidos. Es decir, dada una serie de números determinar si son o no aleatorios.

Pasos para evaluar una prueba de corridas:

- a) Primeramente, le asignaremos un signo a cada número de la secuencia ya sea $+$ o $-$, eso dependerá de los siguiente. Viñetitas
- b) Si a un número le sigue otro mayor, se le asigna $+$. Esto es si $X_i < X_{i+1}$ el signo asignado será $(+)$. Siendo X_i un número de la muestra o secuencia de números.

- c) Si el número siguiente es menor, se le da un signo -. Esto es si $X_i > X_{i+1}$ el signo asignado será (-).
- d) Se continuará con la comparación de los números y la asignación de su signo correspondiente hasta $N-1$. Es decir, hasta el penúltimo número de la secuencia, ya que al último número le sigue un evento nulo (ya no es posible compararlo con otro número) [Ríos & Ríos, 2000].
- e) Una vez encontrado los signos de cada número de la secuencia dada se procede a calcular el total de corridas que resulta de la suma de corrida ascendente con la descendente.
- f) Rechazar la hipótesis de que una secuencia de números es aleatoria, porque hay un número excesivo o demasiado bajo de corridas. Por ende, se requiere una prueba de dos colas para determinar si se ha presentado alguno de esos extremos. Como estadístico de la prueba se utilizará la ecuación 4.

$$Z = \frac{|a - \mu_a|}{\sigma_a} \quad (4)$$

Donde:

a : Número total de corridas en una secuencia

μ_a : La media, ecuación 5.

σ_a^2 : Varianza de a , ecuación 6

$$\mu_a = \frac{2N - 1}{3} \quad (5)$$

$$\sigma_a^2 = \frac{16N - 29}{90} \quad (6)$$

2. Método

En la figura 1 se muestra la metodología de manera esquemática para determinar la calidad de los números aleatorios generados en hojas de cálculo comparando la función: "aleatorio ()" de la hoja de cálculo, y el método de Park-Miller a través de las pruebas de aleatoriedad, de correlación y ji-cuadrada:

- Etapa 1: Establecer la hipótesis nula y alternativa. En esta etapa se examinarán dos hipótesis opuestas: la hipótesis nula y la hipótesis alternativa.

La hipótesis nula es el enunciado que se probará. Por lo general, la hipótesis nula es un enunciado de que "no hay efecto" o "no hay diferencia". La hipótesis alternativa es el enunciado que se desea poder concluir que es verdadero de acuerdo con la evidencia proporcionada por los datos de la muestra. Se utilizará el valor p para tomar esa decisión. Si el valor p es menor que el nivel de significancia (denotado como α o alfa), entonces puede rechazar la hipótesis nula.

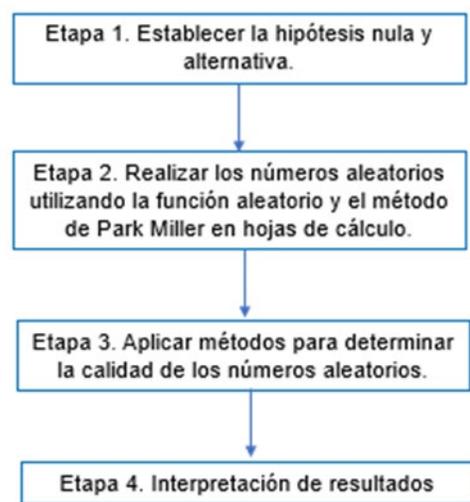


Figura 1 Esquema del método propuesto.

- Etapa 2: Realizar los números aleatorios utilizando la función: "aleatorio ()" y el método de Park Miller en hojas de cálculo. En esta etapa se llevará a cabo la generación de los números aleatorios.

Para Park y Miller se siguieron los siguientes pasos:

- a) Elección de una semilla impar
- b) Un número de inicio de cinco cifras, igualmente, impar
- c) Para el siguiente número, realizar la operación residuo, como se puede ver en ecuación 7.

$$X_1 = \text{Residuo} \left(\frac{\text{Semilla} * \text{Número de inicio}}{2^{31} - 1} \right) \quad (7)$$

- d) Realizar el paso anterior hasta completar mil números. Cabe destacar que el valor de la semilla debe ser siempre el mismo.

- e) En la siguiente columna, partiendo del número que le sigue al de inicio, realizar la operación de la ecuación 8.

$$\frac{\text{Segundo número}}{2^{31} - 1} \quad (8)$$

- f) Replicar el paso anterior hasta completar 1000 números.
- g) Calcular la mediana de los datos calculados en el paso e y f.
- h) Ordenar los datos de menor a mayor
- Etapa 3: Aplicar métodos para determinar la calidad de los números aleatorios:
 - ✓ Ji-cuadrada:
 - a) Realizar una tabla con números del 0.1 al 1.
 - b) Contar la cantidad de números (los ordenados de menor a mayor) que estén debajo de 0.1 con apoyo de la función: "CONTAR.SI ()". Y así sucesivamente hasta completar la tabla.
 - c) En otra columna, en la primera casilla, colocar la cantidad de números que son menores a 0.1. En la segunda casilla, es el resultado de restar la cantidad de números que son menores a 0.2 menos la cantidad de números menores a 0.1. Repetir este paso hasta llegar a la resta de los números menores a 1, menos los menores a 0.9. Esto estima el valor de observados para cada casilla: $(O_i, i=1,10)$
 - d) Obtener el valor esperado E_i . Esto es el resultado de la división de la cantidad de números (1,000) entre 10.
 - e) Usar la fórmula en la ecuación 2. Obtener el valor p.
 - f) Comparar el valor de p con el nivel de significancia de 0.05.
 - ✓ Correlación: Para la prueba de correlación se siguieron varios pasos:
 - a) Con los números previamente generados y ordenados del menor al mayor, hacer cinco columnas idénticas, pero con un renglón de diferencia, de manera que queden en forma de escalera.
 - b) Obtener el coeficiente de correlación de la primera columna vs la segunda. Después, la primera columna vs la tercera y así, sucesivamente.

- c) Obtener los grados de libertad de cada comparación entre columnas, recordando que la fórmula es $n - 2$.
 - d) Calcular el valor de t para la prueba de hipótesis de la correlación de Pearson y calcular el valor de p .
 - e) Concluir si se rechaza o no la hipótesis nula: $\rho = 0$.
- ✓ Prueba de corridas
- a) Calcular la mediana de los números aleatorios que se hayan generado
 - b) Comparar cada número aleatorio con la mediana. Este comparativo busca identificar los números que son menores a la mediana, se identificarán con 1. Mientras que los números mayores a la mediana se identificarán con un 0. Para realizar este paso, se puede utilizar la función: "si ()".
 - c) En otra columna se realizará el paso b), pero ahora con signo contrario. Es decir, si en el paso b) utilizamos el signo menor que, en esta columna se usará el signo mayor que.
 - d) Llamaremos "corridas" a siguiente columna. Se compararán entre ellos mismos los números de la columna generada en paso c).
 - e) Dejando un espacio en blanco, posicionándonos en la segunda casilla y apoyándonos de la función si, llevar a cabo el comparativo, como se muestra en la figura 2.

	A	B	C	D	E
1	datos	0.446398		corridas	
2	0.50434498	0	1		
3	0.93590598	0	1	0	
4	0.99018642	0	1	0	
5	0.2613923	1	0	1	
6	0.36508372	1	0	0	

Figura 2 Utilización de función si, en prueba de corridas.

- f) Se compara el número de abajo con el de arriba. Repetir este paso hasta terminar todas las comparaciones.
 - g) Calcular R media, varianza y valor de z
 - h) Buscar el valor z de tablas con 1-(alfa/2)
 - i) Comparar valores y decidir si H0 se rechaza.
 - j) H0: El orden de los datos es aleatorio
 - k) H1: El orden de los datos no es aleatorio
 - l) Graficar los datos con la mediana obtenida.
- ✓ Prueba de corridas arriba y abajo para números estadísticamente independientes:
- a) Con los números aleatorios previamente generados, y a partir del segundo número de estos, utilizando la función “si”, colocar signos “+” y “- “. Replicar esta acción con todos los números. Se muestra un ejemplo en la figura 3.

B	C	D
Semilla		
48271		
Inicio		
60523		
774022086	0.36043212	
899622800	0.41891951	+
1425352813	0.66373163	+
2124553737	0.98932243	+

Figura 3 Uso de la función condicional “si ()” para obtener signos “+” y “- “.

- b) Seguidamente, empleando nuevamente la función “si”, escribir 0 y 1, según corresponda. Replicar esta acción con todos los números.
- c) Sumar la columna donde se calcularon los números “0” y “1”. Al resultado de la suma se le conocerá como “a”. La figura 4 muestra un resultado.
- d) Calcular μ_a y σ_a^2 con las ecuaciones 5 y 6 de este artículo. Donde la cantidad de números pseudoaleatorios generados, en este caso, es 1000, obteniendo $\mu_a = 666.33$ y $\sigma_a^2 = 177.45$.

B	C	D	E
1494115998	0.695751979	-	1
1382538610	0.643794709	-	0
1319429138	0.614407071	-	0
93917672	0.043733824	-	0
161966295	0.075421433	+	1
1434550865	0.668014803	+	0
1594606900	0.742546702	+	0
1013310479	0.471859462	-	1
275104090	0.128105325	-	0
			641

Figura 4. Suma de la columna de “0” y “1”.

- e) Cuantificar el valor del estadístico Z utilizando la ecuación 4, se observa la sustitución de los datos y el resultado es $|Z| = |1.90|$.
- f) Concluir, rechazando o no rechazando la hipótesis nula.
 H_0 : La secuencia de números es independiente, por lo tanto, la secuencia es aleatoria
 H_1 : La secuencia de números no es independiente, por lo tanto, la secuencia no es aleatoria.
- g) Si el valor absoluto de $|Z|$, calculada es mayor o igual a la Z de las tablas, se rechaza la hipótesis de la independencia de los números.

3. Resultados

Con la prueba de ji-cuadrada, en los números aleatorios generados, tanto por el método de Park- Miller y la función de hoja de cálculo: “ALEATORIO ()”, el valor de p es mayor al nivel de significancia de 0.05. Por lo tanto, H_0 no se rechaza, el valor p (0.8201) y (0.7597). Los datos siguen una distribución uniforme, como se puede apreciar en las figuras 5 y 6:

- H_0 : Los datos se distribuyen de manera uniforme.
- H_1 : Los datos no se distribuyen de una manera uniforme.

La prueba de correlación se corrió, igualmente, con mil números generados por ambos métodos, tal como se muestra en la figura 7.

- H_0 : Correlación=0
- H_1 : Correlación \neq 0

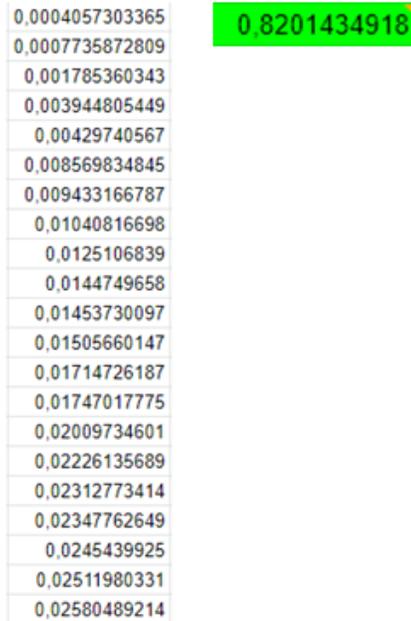


Figura 5 Representación de 1000 números aleatorios y su valor p con la prueba ji-cuadrada.

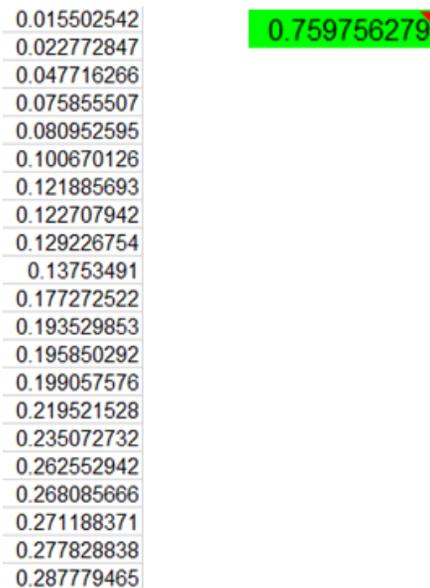


Figura 6 Representación de 1000 números aleatorios con el método de Park- Miller y su valor p con la prueba ji-cuadrada.

	aleatorios				
1	0.015881596				
2	0.970974782	0.0158816			
3	0.36792584	0.97097478	0.0158816		
4	0.008102729	0.36792584	0.97097478	0.0158816	
5	0.329026233	0.00810273	0.36792584	0.97097478	0.0158816
6	0.405053226	0.32902623	0.00810273	0.36792584	0.97097478
7	0.915816655	0.40505323	0.32902623	0.00810273	0.36792584
8	0.878212734	0.91581666	0.40505323	0.32902623	0.00810273
9	0.428020065	0.87821273	0.91581666	0.40505323	0.32902623
10	0.330013419	0.42802007	0.87821273	0.91581666	0.40505323
11	0.313391801	0.33001342	0.42802007	0.87821273	0.91581666
12	0.385166832	0.3133918	0.33001342	0.42802007	0.87821273
13	0.576176043	0.38516683	0.3133918	0.33001342	0.42802007
14	0.608334776	0.57617604	0.38516683	0.3133918	0.33001342
15	0.269098413	0.60833478	0.57617604	0.38516683	0.3133918
16	0.446543381	0.26909841	0.60833478	0.57617604	0.38516683
17	0.965778629	0.44654338	0.26909841	0.60833478	0.57617604
18	0.279335284	0.96577863	0.44654338	0.26909841	0.60833478
19	0.295017126	0.27933528	0.96577863	0.44654338	0.26909841
20	0.834074572	0.29501713	0.27933528	0.96577863	0.44654338
21	0.248619576	0.83407457	0.29501713	0.27933528	0.96577863

Figura 7 Representación de 1000 números aleatorios escalonados para la prueba de correlación.

Se obtuvieron los valores de correlación, mostrados en la tabla 2. Por lo tanto, la correlación se estima tiene un valor de cero.

Tabla 2 Valores de correlación con su valor p.

Pasos	Correlación	GI	T	Valor P	Decisión
1	-0.018569925	997	0.58645221	0.55770442	H ₀ no se rechaza
2	-0.028500516	996	0.89982664	0.36843002	H ₀ no se rechaza
3	-0.006809481	995	0.21480067	0.82996675	H ₀ no se rechaza
4	-0.039140041	994	1.23494433	0.21714305	H ₀ no se rechaza

Se generaron 1000 números aleatorios por el método de Park y Miller, de manera escalonada para la prueba de correlación, tal como se puede ver en la figura 8.

Comparando los valores “p” contra un alfa de 0.05, se puede concluir que la correlación es pequeña, tiene un valor de cero, véase la tabla 3. Por lo tanto, H₀ no se rechaza. Hay ausencia de relación en los números pseudoaleatorios que se calcularon en la figura 9.

Para la prueba de corridas arriba y abajo para números estadísticamente independientes los resultados se obtuvieron que $|Z| = |1.90|$.

0.360432121				
0.418919511	0.360432121			
0.663731626	0.418919511	0.360432121		
0.989322429	0.663731626	0.418919511	0.360432121	
0.582950303	0.989322429	0.663731626	0.418919511	0.360432121
0.594060239	0.582950303	0.989322429	0.663731626	0.418919511
0.881806088	0.594060239	0.582950303	0.989322429	0.663731626
0.66167823	0.881806088	0.594060239	0.582950303	0.989322429
0.869851549	0.66167823	0.881806088	0.594060239	0.582950303
0.604103906	0.869851549	0.66167823	0.881806088	0.594060239
0.699663973	0.604103906	0.869851549	0.66167823	0.881806088
0.47963141	0.699663973	0.604103906	0.869851549	0.66167823
0.287779465	0.47963141	0.699663973	0.604103906	0.869851549
0.402554343	0.287779465	0.47963141	0.699663973	0.604103906
0.700676987	0.402554343	0.287779465	0.47963141	0.699663973
0.378830222	0.700676987	0.402554343	0.287779465	0.47963141
0.513652133	0.378830222	0.700676987	0.402554343	0.287779465
0.502092618	0.513652133	0.378830222	0.700676987	0.402554343
0.512776191	0.502092618	0.513652133	0.378830222	0.700676987
0.219521528	0.512776191	0.502092618	0.513652133	0.378830222
0.52366585	0.219521528	0.512776191	0.502092618	0.513652133
0.874237093	0.52366585	0.219521528	0.512776191	0.502092618
0.298697809	0.874237093	0.52366585	0.219521528	0.512776191
0.441933244	0.298697809	0.874237093	0.52366585	0.219521528
0.559608634	0.441933244	0.298697809	0.874237093	0.52366585

Figura 8 Representación de 1000 números aleatorios por el método de Park y Miller, escalonados para la prueba de correlación.

Tabla 3 Valores de correlación con su valor p.

Pasos	Correlación	GI	T	Valor P	Decisión
1	0.040580105	997	1.282385575	0.200005502	H ₀ no se rechaza
2	0.028244093	996	0.891724303	0.372756095	H ₀ no se rechaza
3	-0.00605283	995	0.019092834	0.019092834	H ₀ no se rechaza
4	-0.05049352	994	1.593981151	0.111258158	H ₀ no se rechaza

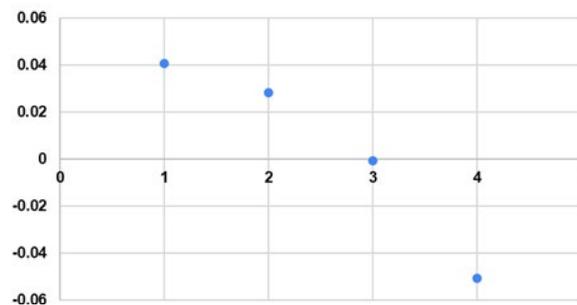


Figura 9 Gráfico de correlaciones de 1000 números aleatorios (en 4 pasos).

Con un valor de $|Z|$ de 1.90 y un valor de Z de tablas de 1.96 (considerando que α es igual a 0.05). Se puede concluir que, con un nivel de confianza de 0.05, H₀ no se rechaza. La secuencia de números es independiente, por lo tanto, la secuencia es aleatoria.

4. Discusión

Como señalan Abasolo, Carranza García, De Luna Pérez, Hernández Martínez & Medrano Meza [Dasso, 2015], los números pseudoaleatorios se generan mediante Algoritmos Determinísticos, divididos en congruenciales y no congruenciales, éstos a su vez se dividen en lineales y no lineales, que requieren parámetros de arranque como una semilla (un valor inicial que se toma como punto de partida para realizar el cálculo del número aleatorio).

En 2005, Bernal, Dolores y Sánchez, realizaron una investigación sobre la aplicación de la simulación con hoja de cálculo a la teoría de colas, donde generan, primeramente, una serie de números pseudoaleatorios, que son validados con la prueba de Kolmogórov-Smirnov, la prueba de la ji-cuadrada y la prueba de los promedios; mientras que para probar la independencia se utiliza el test de rachas y la prueba de póker. Comprobaron que la hoja de cálculo de Excel, siendo una herramienta de tipo ofimático, se revela como una aplicación que permite la simulación de todo tipo de distribuciones estadísticas. En particular, es posible efectuar la simulación de las funciones de distribución propias de la Teoría de Colas, cumpliendo los parámetros obligados de rigor que se debe exigir a este tipo de cálculos. Por otro lado, en un estudio realizado por Sánchez García (2008), se utilizó software de código abierto, el cual, se presenta como una alternativa para realizar todo tipo de simulaciones, obteniendo resultados igual de válidos que con una aplicación comercial como puede ser Microsoft Excel. La generación de números pseudoaleatorios mediante las funciones que incorporan OpenOffice.org Calc y Gnumeric proporciona valores uniformes e independientes que pueden ser aplicados en cualquier técnica de simulación.

5. Conclusiones

El concepto de "aleatoriedad" está muy relacionado con la obtención de resultados inesperados, como vencer a un equipo de fútbol favorito para ganar o la probabilidad de que alguien sea alcanzado por un rayo. Estos eventos, si bien son posibles, son impredecibles y es poco probable que ocurran. Los números deben

generarse de la misma forma, la ocurrencia de uno de ellos no tiene nada que ver con los siguientes.

Con esto se están cumpliendo los objetivos mencionados al inicio. La generación de los números pseudoaleatorios, utilizando el método descrito en el presente artículo sí funciona, puesto que las pruebas con las cuales se evaluó su pseudoaleatoriedad, tuvieron resultados positivos. Cualquier persona, empleando los pasos descritos, puede llevarlo a cabo en cualquier hoja de cálculo, como en un Google sheets, hoja de cálculo de Excel o de Mac. Es decir, son reproducibles. Se elaboró una aplicación para usar en cualquier hoja de cálculo con formato xls, con el fin de que dicha aplicación sea compatible y ejecutable en cualquier sistema operativo sin restricciones de hardware y software, y a su vez, pueda generar secuencias replicables de números aleatorios. Finalmente, el método utilizado es fácil de implementar y editar.

6. Bibliografía y Referencias

- [1] Astaiza, L. [1983]. Los números aleatorios y la ingeniería. Ingeniería e investigación, 7, [7], 55-60.
- [2] Bernal, J., Martínez, S. & Sánchez, J. [2005]. Aplicación de la simulación con hoja de cálculo a la teoría de colas, Electrónica de comunicaciones y trabajos de ASEPUMA, 13, 2-3: https://www.researchgate.net/publication/26440880_Aplicacion_de_la_simulacion_con_hoja_de_calculo_a_la_teor%C3%ADa_de_colas/fulltext/0e60545af0c46d4f0ab07ee7/Aplicacion-de-la-simulacion-con-hoja-de-calculo-a-la-teoria-de-colas.pdf.
- [3] Coss Bú, R. [2003]. Simulación: Un enfoque práctico. [1ra ed]. Ciudad de México: Limusa.
- [4] Dasso, D. & Mendoza, D. [2015]. Calidad de los generadores de números pseudoaleatorios, ¡Estadísticas en acción!, 3: http://redi.exactas.unlpam.edu.ar/xmlui/bitstream/handle/2013/166/EeA_Reportes2015.pdf?sequence=1.
- [5] Lehmer, D. [1951]. Proceedings of 2nd Symposium on Large-Scale Digital Calculating Machinery. Mathematical methods in large-scale computing units. Harvard University Press, 26, 141-146.

- [6] Dennis, T. & Johnson, S. [2007]. *Cryptography for developers*. [1ra ed]. Washington: Syngress. 91-137.
- [7] Kendall, M. & Babington, S. [1938]. Aleatoriedad y números de muestreo aleatorio. *Royal Statistical Society*, 1, 147-166.
- [8] Knuth, D. [1998]. *The Art of Computer Programming*. [3ra ed]. Addison Wesley Longman.
- [9] Marsaglia, G. [1993]. Technical correspondence: Remarks on Choosing and Implementing Random Number Generators. *Communications of the ACM*, 36, [7], 105-108.
- [10] Park, S., Miller, W. & Stockmeyer, P. [1988]. Technical correspondence: Response. *Communications of the ACM*. 7, [36],108-110.
- [11] Park, S. & Miller, W. [1988]. Random Number Generators: Good ones are hard to find. *Communications of the ACM*, 31, 1192-1201.
- [12] Payne, W., & Rabung, J. [1969]. Coding the Lehmer pseudo-random number generator. *Communications of the ACM*. 2, 85-86.
- [13] Ríos, D., & Ríos, S. [2000]. *Simulación: Métodos y aplicaciones*. [2da ed]. Ciudad de México: Alfaomega.
- [14] Sánchez, J. [2008]. Simulación con hojas de cálculo de código abierto: OpenOffice. org y Gnumeric. *Encuentro internacional Rect@s*, 1, 1-11.
- [15] Sullivan, S. [1993]. Technical correspondence: Another test for randomness. *Communications of the ACM*, 7,[36], 108.
- [16] Von Neumann, J. [1951]. Various techniques used in connection with random digits. *National Bureau of Standards Applied Mathematics Series*, 12, 36-38.
- [17] Wichmann, B., & Hill, D. [1982]. Algorithm AS 183: An Efficient and Portable Pseudo-Random Number Generator. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 2, 188-190.
- [18] Widynski, B. (2020). Middle Square Weyl Sequence RNG. *Arxiv*, 1, 1-14.