TP 2.1 GENERADORES PSEUDOALEATORIOS

Ornella Colazo
Universidad Tecnológica Nacional
Ingeniería en Sistemas
Legajo 47864
ornecolazo@gmail.com

Diego Navarro
Universidad Tecnológica Nacional
Ingeniería en Sistemas
Legajo 48029
navarrodiego201513@gmail.com

Matias Petrich
Universidad Tecnológica Nacional
Ingeniería en Sistemas
Legajo 46852
matias.petrich@gmail.com

Ramiro Cordoba
Universidad Tecnológica Nacional
Ingeniería en Sistemas
Legajo 46824
ramirocordobautn@gmail.com

Matias Ferullo
Universidad Tecnológica Nacional
Ingeniería en Sistemas
Legajo 48039
matias.ferullo1@gmail.com

18 de Abril, 2023

ABSTRACT

En la naturaleza se presentan una gran cantidad de sucesos que resultan impredecibles. Para tratar de obtener una mayor certidumbre en estos casos, se utiliza la simulación como herramienta. La simulación es una técnica numérica que se basa en llevar a cabo experimentos matemáticos y lógicos en una computadora, para lo cual es necesario crear un modelo del sistema que se está analizando. La finalidad de estos experimentos es, principalmente, comprender los comportamientos del sistema o evaluar diversas estrategias de funcionamiento del mismo.

Dentro de la simulación por computadora, los llamados "números aleatorios" son una pieza clave, ya que son ellos los que determinan qué suceso impredecible se produce o no en la simulación. Estos números son la representación de la incertidumbre en el modelo del sistema. Sin embargo, es importante cuestionarse ¿qué tan aleatorios son estos números? y ¿de dónde provienen?. En este estudio, realizamos los generadores Lineal Congruencial, Lineal Congruencial Multiplicativo, método de Cuadrados Medios y el generador de Python. A cada generador se le realizo un test de Poker, de Corridas y de Chi Cuadrado.

1 Introducción.

El hombre siempre se ha interesado por el azar, el futuro y el destino. Es por esto que se ha dedicado mucho tiempo al estudio de la aleatoriedad. Gracias a esto, la teoria de la probabilidad ha sido constantemente desarrollada y hoy en día es aplicada en diversos campos de estudio, siendo la computación uno de ellos.

En 1945, Von Neumann, uno de los pioneros del campo de los números aleatorios, previó el gran potencial de las computadoras para resolver problemas estocásticos. En un problema estocástico las variables del evento se comportan aleatoriamente. El auge y creciente uso de estas teorías se debe a la utilización de números aleatorios generados en computadoras.

Hoy en día, los ingenieros nos enfrentamos a problemas donde es imposible obtener una respuesta analítica. Debido a esto, la implementación de de la simulación en computadoras han ayudado a resolver estos problemas.

La simulación es una técnica que requiere de procedimientos capaces de producir números aleatorios, ya que a la hora de estudiar un evento estocástico, como hemos explicado antes, las variables se comportan aleatoriamente.

Los números aleatorios son aquellos que pueden ser generados a partir de fuentes de aleatoriedad, generalmente, la naturaleza y son gobernados por las leyes del azar. Podemos entender por número aleatorio a aquel que puede ser

generado con igual probabilidad y en forma independiente de cualquier resultado previo. Estadisticamente, significa que los números aleatorios son variables aleatorias, independientes y con distribución uniforme.

Además, los números aleatorios son la base de la simulación. Un generador de números aleatorios produce una sucesión de valores que supuestamente son realizaciones de una secuencia de variables aleatorias independientes e idénticamente distribuidas (i.i.d.) U(0,1).

Un buen generador de números aleatorios debe cumplir con algunas propiedades. Los números producidos deben parecer como obtenidos de una distribución uniforme U(0,1). En segundo lugar, los números deben ser independientes.

Como características del generador, este debe ser rápido, breve, no degenerativo, es decir que no genere continuamente el mismo número, y funcionar por un período largo antes de repetir la secuencia.

El método más conveniente de generar números aleatorios es utilizar algoritmos que posean alguna base matemática. Estos algoritmos producen una sucesión de números que se asemeja a la de una sucesión de realizaciones de variables aleatorias iid U(0,1), aunque realmente no lo sea. Es por ello que este tipo de números se denominan pseudoaleatorios y el algoritmo que los produce se llama generador de números pseudoaleatorios.

Los números pseudoaleaorios no son estrictamente aleatorios porque su generación es determinística. Es decir, que si conocemos la semilla y el algoritmo a implementar, los números serán predecibles.

El objetivo de los generadores de números pseudoaleatorios (PRNG, por sus siglas en inglés) es obtener secuencias que se comporten como si fueran aleatorias. La salida de estas sequencias son indistinguibles estadísticamente, por lo que los números de los PRNG suelen parecer más random que los obtenidos de generadores de números aleatorios (TRNG, por sus siglas en inglés)

2 Generadores.

Un generador de números aleatorios es un programa que entrega una serie de bits aleatorios, es decir, impredecibles desde el punto de vista externo. Estos bits se pueden usar para crear un número aleatorio.

Existen generadores de diferentes tipos dependiendo de su fuente de entropía (información impredecible) y de cómo la usen.

Generador de números Aleatorios TRNG En primer lugar, tenemos los generadores que se conocen como generadores de números aleatorios (TRNG). Los TRNG obtienen fuentes de entropía de lugares físicas, porque el mundo real es impredecible. Miden los cambios en los semiconductores, la forma en que mueve el mouse, el teclado, la información de los sensores de la computadora, los micrófonos, las redes y muchas otras cosas. Algunos incluso cosechan entropía de procesos cuánticos, llamados generadores cuánticos de números aleatorios o QRNG.

Estos elementos físicos son una fuente confiable de entropía, pero no podemos confiar en la forma en que los capturamos ya que pueden ser tergiversados por usuarios malintencionados. También son lentos para generar bits aleatorios que una aplicación podría necesitar.

Es normal que un TRNG use suficiente entropía para servir a las aplicaciones que lo usan y que eventualmente se bloquee o se vuelva inseguro.

Debidos a los problemas que pueden tener los TRNG, estudiaremos los generadores de número pseudoaleatorios.

2.1 Generadores de número Pseudoaleatorios PRNG

Un generador de numeros pseudoaleatorios es una estructura G = (X, x0, T, U, g), donde X es un conjunto finito de estados, x0 $\in$ X es el estado inicial (semilla), la aplicacion T : X $\rightarrow$ X es la funcion de transicion, U es el conjunto finito de posibles observaciones, y G : X $\rightarrow$ U es la funcion de salida.

Básicamente, el funcionamiento de un generador de números pseudoaleatorios consiste en elegir una semilla inicial cualquiera x_0, y se genera una sucesión de valores x_n mediante una relación de recurrencia.

$$x_n = T(x_{n-1}) \tag{1}$$

Cada uno de estos valores proporciona un número pseudoaleatorio u_n definido a través de alguna relación $u_n = g(x_n)$. La sucesión de estados es periódica, puesto que X es finito. En algún momento, ocurrirá que $x_j = x_i$ para algún $j > i$, y a partir de ese instante, $x_{j+k} = x_{i+k}$, y por lo tanto, $u_{j+k} = u_{i+k}$, para todo $k \geq 0$. El período es el menor entero $\rho > 0$ tal que para algún entero $\tau \geq 0$, se verifica que $x_{\rho+k} = x_k$, para todo $k \geq \tau$. Claramente, el período de

un generador no puede exceder el cardinal del espacio de estados. Lo que significa que el periodo máximo posible que puede tener un generador está limitado por el número de valores posibles en el conjunto de estados. Una buena propiedad para un generador es que su periodo esté cercano a $|X|$.

Un buen generador de números pseudoaleatorios debería tener las siguientes propiedades:

- Por encima de todo, la sucesión de valores que proporcione debería asemejarse a una sucesión de realizaciones independientes de una variable aleatoria U(0, 1)
- Los resultados deben ser reproducibles, en el sentido de que comenzando con las mismas condiciones iniciales debe ser capaz de reproducir la misma sucesión. Esto nos puede permitir depurar fallos del modelo o simular diferentes alternativas del modelo en las mismas condiciones obteniendo una comparación más precisa. Los procedimientos físicos no permiten que los resultados sean reproducibles.
- La sucesión de valores generados debe tener un ciclo no repetitivo tan largo como sea posible
- El generador debe ser rápido y ocupar poca memoria interna

2.1.1 Generador Congruente Lineal

Un generador lineal congruencial (GLC) es un algoritmo que permite obtener una secuencia de números pseudoaleatorios calculados con una función lineal definida a trozos discontinua. Es uno de los métodos más antiguos y conocidos para la generación de números pseudoaleatorios.

Se llaman congruentes porque un número entero x es congruente con otro número y, de módulo m, si x-y es divisible por m.

En los generadores congruenciales lineales se considera una combinación lineal de los últimos k enteros generados y se calcula su resto al dividir por un entero fijo m. En el método congruencial simple (de orden $k = 1$), partiendo de una semilla inicial x_0, el algoritmo secuencial es el siguiente:

$$X_i = (aX_{i-1} + C) \bmod m \tag{2}$$

donde a (multiplicador), c (incremento) y m (módulo) son enteros positivos fijados de antemano (los parámetros de este generador). Si $c = 0$ el generador se denomina congruencial multiplicativo. En caso contrario, se llama mixto.

Obviamente los parámetros y la semilla determinan los valores generados. Cada entero X_i queda completamente determinado por las constantes m, a, c y X_0, entonces puede demostrarse por inducción matemática que para i=1,2,3,...:

$$X_i = (a^i X_0 + c\frac{a^i - 1}{a - 1}) \bmod m \tag{3}$$

En las fórmulas utilizadas a es un multiplicador, m el módulo, C el incremento y X_0 es la semilla (o valor inicial), y deben satisfacer:

$$a, 0 < a < m; \tag{4}$$

$$m, O < m; \tag{5}$$

$$C, 0 \leq C < m; \tag{6}$$

$$X_0, 0 \leq X_0 < m \tag{7}$$

Aunque los GLC son capaces de producir números pseudoaleatorios que pueden pasar las pruebas de aleatoriedad, esta posibilidad es extremadamente sensible a la elección de los parámetros m, a y c

Algunas recomendaciones de los valores a elegir:

- Semilla: la semilla inicial debe ser elegida de manera aleatoria y debe ser diferente para cada ejecución del generador. Se recomienda utilizar una semilla lo suficientemente grande para evitar que se repita la secuencia generada.
- Multiplicador: el multiplicador debe ser elegido de manera que tenga un periodo máximo y que proporcione una buena distribución de los números generados. Los valores típicos incluyen números primos grandes o números con un gran factor primo.

- Incremento: el incremento también debe ser elegido para proporcionar un buen periodo máximo y una buena distribución. Los valores típicos incluyen valores impares o valores que son coprimos con el módulo.
- Módulo: el módulo debe ser lo suficientemente grande para evitar ciclos cortos en la secuencia generada. Se recomienda usar un número primo o un número que tenga factores primos grandes.

Teorema A: Un generador congruencial tiene período completo si y solo si se cumplen las siguientes condiciones (q=m):

1. m y b son primos entre sı
2. Si q es un número primo que divide a m, entonces q divide a a-1.
3. Si 4 divide a m, entonces 4 divide a a-1.

Algunas consecuencias:

- Si m primo, p=m si y solo si a =1.
- Un generador multiplicativo no cumple la condición 1 (m.c.d.(0,m)=m).

2.1.2 Generador congruente multiplicativo

Los generadores congruenciales multiplicativos son un tipo de algoritmo utilizado para generar números pseudoaleatorios en una secuencia determinada. Se basan en la fórmula $X_{n+1} = (a \cdot X_n) \mod m$, donde X_n es el número generado en la posición n de la secuencia, a es un número entero positivo llamado multiplicador, y m es un número entero positivo llamado módulo.

Es importante destacar que un generador congruencial multiplicativo no puede tener un periodo completo, es decir, no puede generar todos los posibles números pseudoaleatorios sin repetirse. Sin embargo, es posible obtener un periodo muy largo si se eligen adecuadamente los valores de a y m. En general, se considera que los generadores multiplicativos son mejores que los aditivos, ya que no requieren de la adición de una constante c, lo que simplifica la implementación.

Para elegir los valores adecuados de a y m, es importante considerar ciertas propiedades estadísticas que deben cumplir los números generados, como la uniformidad, independencia y aleatoriedad. También es necesario elegir cuidadosamente la semilla X_0, que es el valor inicial de la secuencia.

Una elección adecuada de m es que sea igual a 2^k, donde k es el tamaño de palabra del microprocesador, lo que permite aprovechar el desbordamiento de datos y optimizar el tiempo de cómputo. Además, se recomienda que el valor de a sea un número primo con respecto a m, y que sea grande en comparación con m, para obtener una secuencia más aleatoria.

En conclusión, los generadores congruenciales multiplicativos son una herramienta útil para generar secuencias de números pseudoaleatorios, aunque es importante elegir adecuadamente los valores de a, m y la semilla X_0 para obtener una secuencia que cumpla con las propiedades estadísticas requeridas y sea lo más aleatoria posible.

2.1.3 Método de los Cuadrados Medios

Este método se debe fue propuesto en los años 40 por los matemáticos John Von Neumann y Nicholas Metrópolis. El método comienza tomando un número al azar, x_0, de $2n$ cifras (originalmente los autores proponían 4 cifras) que al elevarlo al cuadrado resulta un número de hasta $4n$ cifras. Si es necesario se añaden ceros a la izquierda para que el número resultante tenga exactamente $4n$ cifras. Sea x_1 el número resultante de seleccionar las $2n$ cifras centrales de x_0^2 ; el primer número aleatorio u_1 se obtiene poniendo un punto decimal delante las $2n$ cifras de x_1. A continuación x_2 y u_2 se generan a partir de x_1 del mismo modo. Así sucesivamente.

Este método tiene dos inconvenientes principales:

- tiene una fuerte tendencia a degenerar a cero rápidamente (probar por ejemplo con $x_0 = 1009$)
- los números generados pueden repetirse cíclicamente después de una secuencia corta.

Ejemplo:

$$x_0 = 3708 \Rightarrow x_0^2 = 13|7492|64 \Rightarrow x_1 = 7492 \Rightarrow u_1 = 0.7492$$
$$x_1 = 7492 \Rightarrow x_1^2 = 56|1300|64 \Rightarrow x_2 = 1300 \Rightarrow u_2 = 0.1300$$
$$x_2 = 1300 \Rightarrow x_2^2 = 1|6900|00 \Rightarrow x_3 = 6900 \Rightarrow u_3 = 0.6900$$
$$x_3 = 6900 \Rightarrow x_3^2 = 47|6100|00 \Rightarrow x_4 = 6100 \Rightarrow u_4 = 0.6100$$
$$x_4 = 6100 \Rightarrow x_4^2 = 47|2100|00 \Rightarrow x_5 = 2100 \Rightarrow u_5 = 0.2100$$

3 Tests

Hay dos tipos de pruebas:

- Empíricas: evalúan estadísticas de sucesiones de números.
- Teóricas: se establecen las características de las sucesiones usando métodos de teoría de números con base en la regla de recurrencia que generó la sucesión.

Realizaremos pruebas empíricas.

3.1 Test de aleatoriedad de Chi-Cuadrado

El test de aleatoriedad de Chi-Cuadrado es una prueba de hipotesis estadistica empirica que indica si una muestra representa (o se asemeja) razonablemente los datos esperados a encontrar en una población, analizando su uniformidad. Es decir, Se trata de decidir si los números generados se pueden considerar como una realización de una muestra aleatoria simple de una distribución $U(0, 1)$.

Prueba: Dada la muestra u_1, ..., u_n y un nivel de significación α, el test consiste en los siguientes pasos:

1. Dividir el intervalo (0, 1) en k clases disjuntas de la misma amplitud, $1/k$. Para cada clase C_j , contar el número de elementos O_j que cae en dicha clase.
2. Comparamos las frecuencias observadas en cada clase con las que corresponderían según la distribución teórica. Se considera el estadístico:
$$T = \sum_{j=1}^{k} \frac{(O_j - \frac{n}{k})^2}{\frac{n}{k}} \tag{8}$$
Se demuestra que para n grande, T sigue una distribución χ^2 con $k - 1$ grados de libertad.
3. Se rechaza la hipótesis de uniformidad si $T > \chi^2_{k-1,\alpha}$, donde $\chi^2_{k-1,\alpha}$ es el percentil de orden $1 - \alpha$ de la distribución χ^2_{k-1}.

3.2 Test de Poker

El test de póker es una prueba de aleatoriedad que se utiliza para verificar la calidad de los generadores de números aleatorios. Esta prueba consiste en visualizar el número r_i con cinco decimales (como si fuera una mano del juego de póker, con cinco cartas), y clasificarlo como:

- TD: todos diferentes.
- 1P: exactamente 1 par.
- 2P: dos pares.
- T: una terna (3 dígitos iguales).
- TP: una terna y un par.
- P: póker (4 dígitos iguales).
- TI: Todos iguales.

A modo de aclarar mejor la clasificación, se muestra a continuación algunos ejemplos junto con su clasificación correspondiente:
r_i=0.69651 se le clasifica como par, porque hay dos números seis.
r_i=0.13031, se le clasifica como dos pares (dos números unos y dos números tres).
r_i=0.98898 se le clasifica como una tercia y un par, porque hay tres números ochos y dos números nueve.

La prueba póker puede realizarse a números $r_i$con **tres, cuatro y cinco decimales**. Dependiendo de los decimales que se van a tener en cuenta, las categorías se reducen o expanden.

Categoria	Probabilidad observadas (O_i)	Probabilidad esperada (E_i)
Todos diferentes (TD)	0,72	0,72n
Exactamente un par (1P)	0,27	0,27n
Terna (T)	0,01	0,01n

Table 1: Prueba de Poker con tres decimales

.

Categoria	Probabilidad observadas (O_i)	Probabilidad esperada (E_i)
Todos diferentes (TD)	0,5040	0,5040n
Exactamente un par (1P)	0,4320	0,4320n
Dos pares (2P)	0,0270	0,0270n
Terna (T)	0,0360	0,0360n
Poker (P)	0,0010	0,0010n

Table 2: Prueba de Poker con cuatro decimales

.

Categoria	Probabilidad observadas (O_i)	Probabilidad esperada (E_i)
Todos diferentes (TD)	0,3024	0,3024n
Exactamente un par (1P)	0,5040	0,5040n
Dos pares (2P)	0,1080	0,1080n
Una terna y un par (TP)	0,0090	0,0090n
Terna (T)	0,0720n	0,0720nn
Poker (P)	0,0045	0,0045n
Todos iguales (TI)	0,0001	0,0001n

Table 3: Prueba de Poker con cinco decimales

.

La prueba póker requiere el estadístico de la distribución Chi-cuadrada $X^2_{\alpha,6}$ para números con cinco decimales, $X^2_{\alpha,4}$ para números con cuatro decimales y $X^2_{\alpha,2}$ para números con tres decimales.

El procedimiento de la prueba consiste en:

1. Determinar la categoría de cada número del conjunto r_i.
2. Contabilizar los números r_i de la misma categoría o clase para obtener las frecuencias observadas (O_i).
3. Calcular el estadístico de la prueba $(X_0)^2$ con la ecuacion

$$X_0^2 = \sum_{i=1}^{m} \frac{(E_i - O_i)^2}{E_i} \tag{9}$$

   Donde E_i es la frecuencia esperada de los números r_i en cada categoría y m representa la cantidad de categorías para la prueba póker con cinco, cuatro y tres decimales, respectivamente.
4. Finalmente, podemos comparar la distribución real de combinaciones en una secuencia de números aleatorios con la distribución esperada calculada. Si la distribución real se desvía significativamente de la distribución esperada, entonces podemos concluir que el generador de números aleatorios no es de alta calidad y necesita ser mejorado.

Frecuencia esperada (E_i):
A continuación, se explica como calcular la frecuencia esperada para el caso de cuatro decimales, ya que el mismo fue el que se tomo en cuenta para este estudio.
Para calcular la frecuencia esperada (E_i), podemos utilizar la fórmula de probabilidad de una distribución uniforme, que es:

$$P(A) = \frac{1}{n} \tag{10}$$

Donde P(A) es la probabilidad de que ocurra un evento A y n es el número total de eventos posibles. En nuestro caso, podemos considerar que cada combinación de cuatro dígitos es un evento posible, por lo que n = 10000.
A partir de esta fórmula, podemos calcular la probabilidad de que ocurra cada tipo de combinación. Por ejemplo, la probabilidad de que aparezca un número específico de cuatro dígitos es $\frac{1}{10000}$, la probabilidad de que aparezcan dos números iguales y dos diferentes es $\frac{1099}{10000}$, la probabilidad de que aparezcan tres números iguales y uno diferente es $\frac{10*9}{1000}$, y así sucesivamente.
Una vez que hemos calculado la probabilidad de cada tipo de combinación, podemos proceder a calcular la frecuencia esperada de cada uno de ellos. La frecuencia esperada se refiere al número de veces que esperamos que aparezca un determinado tipo de combinación en una secuencia de números aleatorios. Para calcular la frecuencia esperada, simplemente multiplicamos la probabilidad de cada tipo de combinación por el número total de eventos en la secuencia. Por ejemplo, si tenemos una secuencia de 1,000 números aleatorios, la frecuencia esperada de un número específico de cuatro dígitos es $(\frac{1}{10000} * 1,000 = 0.1)$, es decir, esperamos que aparezca una vez en la secuencia.

3.3 Test de Runs - Arriba y abajo

El procedimiento de esta prueba consiste en determinar una secuencia de números (S) que solo contiene unos y ceros, de acuerdo con una comparación entre r_i y r_{i-1}. Posteriormente se determina el número de corridas observadas C_0 (una corrida se identifica como la cantidad de unos o ceros consecutivos). Luego se calcula el valor esperado, la varianza del número de corridas y el estadístico Z_0, mediante las ecuaciones.

Media o valor esperado del número de corridas es:

$$\mu_{C_0} = \frac{2n-1}{3} \tag{11}$$

Varianza del número de corridas es:

$$\sigma^2_{C_0} = \frac{16n-29}{90} \tag{12}$$

El estadístico de prueba de la distribución normal es:

$$Z_0 = \left| \frac{C_0 - \mu_{C_0}}{\sigma_{C_0}} \right| \tag{13}$$

Si el estadístico Z_0 es mayor que el valor $Z_{\alpha/2}$, se concluye que los números del conjunto r_i no son independientes. De lo contrario no se puede rechazar que el conjunto r_i sea independiente.

Ejemplo cómo se aplica la prueba de corridas o rachas (Runs Test).

Realizar la prueba de Corridas arriba y abajo a un nivel de aceptación de 95% al siguiente conjunto r_i.

0.34	0.83	0.96	0.47	0.79	0.99	0.37	0.72	0.06	0.18
0.67	0.62	0.05	0.49	0.59	0.42	0.05	0.02	0.74	0.67
0.46	0.22	0.99	0.78	0.39	0.18	0.75	0.73	0.79	0.29
0.11	0.19	0.58	0.34	0.42	0.37	0.31	0.73	0.74	0.21

Table 4: Ejemplo Test de Runs

Realizamos la asignación de unos y ceros por filas. Por tanto la secuencia S es.

S=1,1,0,1,1,0,1,0,1,1,0,0,1,1,0,0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,1,1,0,1,0,0,1,1,0

Se utiliza la siguiente regla para asignar los valores binarios: Si un número es mayor o igual que su predecesor, se le asigna un 1. Si un número es menor que su predecesor, se le asigna un 0.

Obteniéndose un valor de n=40, $C_0 = 24$ y $\alpha = 0.05$. A continuación se presentan los cálculos correspondientes al valor esperado y la varianza del número de corridas:

$$\mu_{C_0} = \frac{2n-1}{3} = \frac{2*40-1}{3} = 26.333 \tag{14}$$

$$\sigma^2_{C_0} = \frac{16n - 29}{90} = \frac{16 * 40 - 29}{90} = 6.788 \tag{15}$$

$$\sigma_{C_0} = 2.605 \tag{16}$$

3.4 Test de Runs - Arriba y abajo de la media

El test de runs arriba y abajo de la media es una prueba estadística que se utiliza para determinar si una secuencia de datos numéricos es aleatoria o si hay algún patrón significativo en los datos. Esta prueba es una extensión de la prueba de rachas o runs, que se utiliza para datos binarios (por ejemplo, una secuencia de "éxitos" y fracasos" en un experimento).

Se parte de la suposición de que los datos provienen de una población con una distribución normal. La prueba se basa en contar el número de rachas "o runs" en la secuencia de datos que están por encima de la media y el número de rachas que están por debajo de la media.

Para llevar a cabo la prueba, primero se calcula la media de la muestra y se divide la secuencia de datos en dos partes: una parte de los datos que son mayores o iguales a la media y otra parte de los datos que son menores a la media. Luego, se cuentan las rachas que hay en cada una de estas dos partes y se comparan con el número esperado de rachas que se obtendría si los datos fueran aleatorios.

Para una secuencia de tamaño N, si n_1 observaciones están por encima de la media y n_2 observaciones están por debajo de la media, entonces el número total de corridas b puede calcularse utilizando la siguiente fórmula:

$$\mu_b = +1 + \sum_{k=1}^{K} n_k \tag{17}$$

donde K es el número de corridas y n_k es la longitud de la k-ésima corrida.

Para evaluar si el número de corridas en una secuencia es significativamente diferente de lo que se esperaría en una secuencia aleatoria, se puede utilizar la prueba de corridas. Se pueden calcular la media y la varianza de b usando las siguientes fórmulas:

$$b = \frac{2n_1n_2}{N} + 1 \tag{18}$$

$$\sigma^2_b = \frac{2n_1n_2(2n_1n_2 - N)}{(N^2)(N-1)} \tag{19}$$

Si n_1 o n_2 son mayores que 20, se puede aproximar la distribución de b como una distribución normal. Entonces, se puede calcular un valor Z para la muestra utilizando la siguiente fórmula:

$$Z_0 = \frac{b - \frac{2n_1n_2}{N} - \frac{1}{2}}{\sqrt{\frac{2n_1n_2(2n_1n_2-N)}{N^2(N-1)}}} \tag{20}$$

Si Z_0 cae dentro del intervalo crítico definido por los valores $z_{\alpha/2}$, donde α es el nivel de significación y $z_{\alpha/2}$ es el valor crítico de la distribución normal estándar correspondiente al nivel de significación, entonces se puede aceptar la hipótesis nula de que la secuencia es aleatoria. De lo contrario, se rechaza la hipótesis nula y se concluye que la secuencia exhibe algún tipo de patrón o tendencia.

Una vez calculado el número esperado de rachas, se puede utilizar una prueba de hipótesis para determinar si el número observado de rachas es significativamente diferente del número esperado. En general, si el valor observado de rachas está muy por debajo o por encima del valor esperado, se puede concluir que la secuencia de datos no es aleatoria y que hay algún patrón significativo en los datos.

4 Conclusiones.

4.1 Estudio del 'Cuadrados Medios'

Para este método, decidimos variar la semilla 2 veces. Estos fueron los resultados

TEST	CHI CUADRADO	POKER	RUN ARRIBA Y ABAJO	RUN ARRIBA Y ABAJO DE LA MEDIA
RESULTADO PARA S= 3708	APROBADO	NO APROBADO	NO APROBADO	NO APROBADO
RESULTADO PARA S= 1908	NO APROBADO	NO APROBADO	NO APROBADO	APROBADO

Figure 1: Tabla de resultados de la semilla S=3708 y S=1908

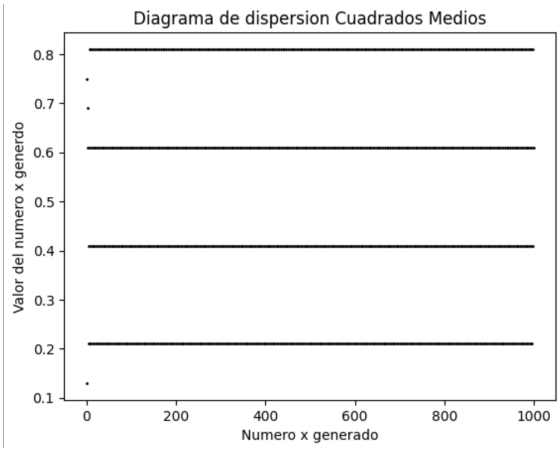

Figure 2: Diagrama de Dispersión de Cuadrados Medios con S= 3708

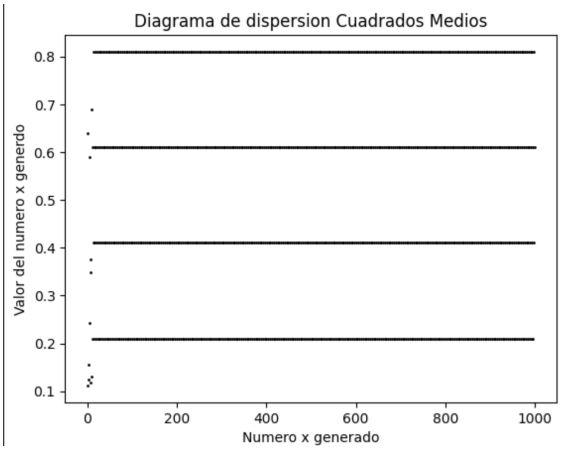

Figure 3: Diagrama de Dispersión Cuadrados Medios con S= 1908

Se puede observar como se generan lineas en el Diagrama de dispersión, lo que significa que los números no son realmente pseudo-aleatorios, si no que se genera un patrón. Además, no aprueba 3/4 tests. Es decir que no es un buen generador pseudoaleatorio.

4.2 Estudio del 'GCL Mixto'

Para este generador, corrimos los 4 tests para 4 parámetros distintos. Las semillas las elegimos al azar.

Primero para los parámetros de GLIBC, la biblioteca de C de GNU, es la biblioteca de tiempo de ejecución estándar del lenguaje C de GNU. Luego con los parámetros de MMIX, es una arquitectura de 64 bits, provista de 256 registros de propósito general de tipo RISC y 32 registros de 64 bits de propósito especial. También con los parámetros de Random en Java. Y por último, modificamos los parámetros de Java, dejando el multiplicador igual, para variar los resultados. Estos fueron nuestros resultados.

PARÁMETROS	TEST	CHI CUADRADO	POKER	RUN ARRIBA Y ABAJO	RUN ARRIBA Y ABAJO DE LA MEDIA
GLIBC	RESULTADO PARA LOS VALORES: s = 12345 mod =2**31 - 1 mult = 1103515245 inc = 12345	APROBADO	NO APROBADO	APROBADO	NO APROBADO
MMIX by Donald Knuth	RESULTADO PARA LOS VALORES: s= 12345 mod =2**64 mult = 6364136223846793005 inc = 1442695040888963407	APROBADO	NO APROBADO	APROBADO	APROBADO
JAVA	RESULTADO PARA LOS VALORES s = 12345 mod =134456 mult = 25214903917 inc = 11	NO APROBADO	NO APROBADO	NO APROBADO	NO APROBADO
Elaboración propia	RESULRADO PARA LOS VALORES s = 54123 mod =2**30 mult = 25214903917 inc = 1442695040888963407	APROBADO	NO APROBADO	APROBADO	APROBADO

Figure 4: Tabla de Restulados de GCL Mixto

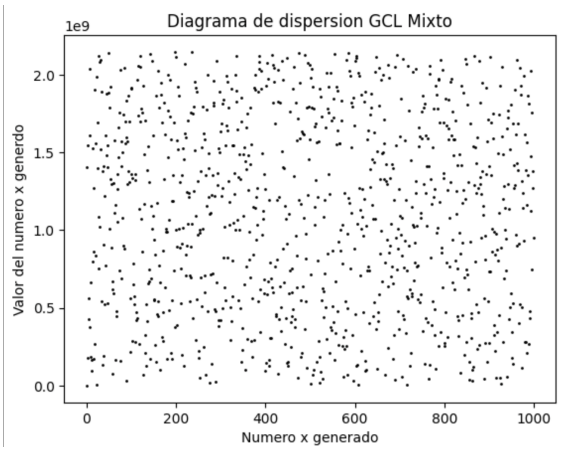

Figure 5: Diagrama de Dispersión GCL Mixto con parámetros glibc

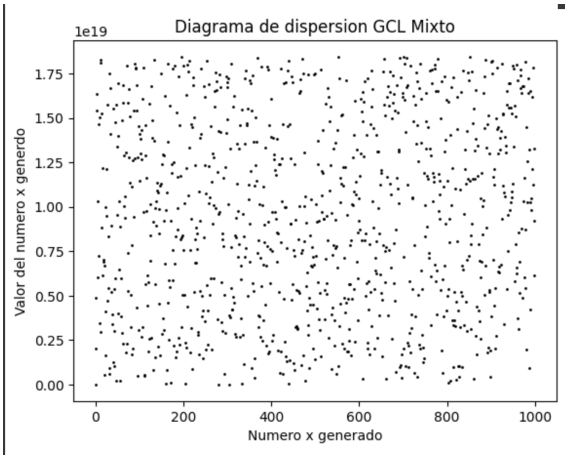

Figure 6: Diagrama de Dispersión GCL Mixto con parámetros de Knuth

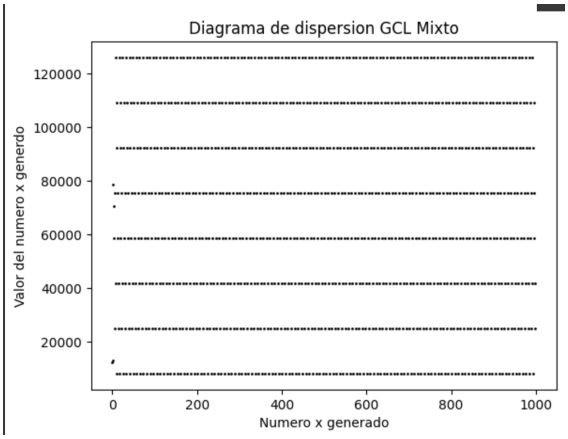

Figure 7: Diagrama de Dispersión GCL Mixto con parámetros de Java

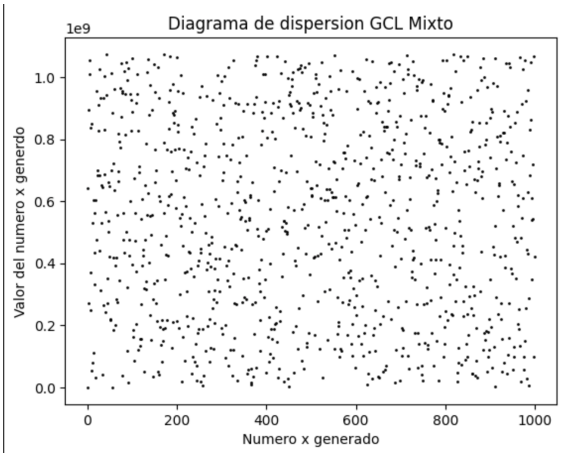

Figure 8: Diagrama de Dispersión GCL Mixto con parámetros elaboración propia

En este estudio, variamos los parámetros 4 veces para obtener distintos resultados. Con los parámetros de Knuth y los que elaborammos nosotros aprueban 3/4 tests y el de glibc, 2/4. Sus gráficas de dispersión no genera patrones, por lo que es un buen generador de números pseudoaleatorios. Sin embargo, cuando probamos los parámetros de Java

encontramos que, por más que es un generador bueno y confiable, con los parámetros equivocados podría generar patrones y dejar de ser un buen generador.

4.3 Estudio del 'GCL Multiplicativo'

Se aplicaron los mismos parámetros que para el GCL Mixto, con las mismas consideraciones.

PARÁMETROS	TEST	CHI CUADRADO	POKER	RUN ARRIBA Y ABAJO	RUN ARRIBA Y ABAJO DE LA MEDIA
GLIBC	RESULTADO PARA LOS VALORES: s = 12345 mod =2**31 - 1 mult = 1103515245 inc = 12345	APROBADO	NO APROBADO	APROBADO	APROBADO
MMIX by Donald Knuth	RESULTADO PARA LOS VALORES: s= 12345 mod =2**64 mult = 6364136223846793005 inc = 1442695040888963407	APROBADO	NO APROBADO	APROBADO	APROBADO
JAVA	RESULTADO PARA LOS VALORES s = 12345 mod =134456 mult = 25214903917 inc = 11	NO APROBADO	NO APROBADO	NO APROBADO	NO APROBADO
Elaboracón propia	RESULRADO PARA LOS VALORES s = 54123 mod =2**30 mult = 25214903917 inc = 1442695040888963407	APROBADO	NO APROBADO	APROBADO	APROBADO

Figure 9: Tablade de Resultados de GCL Multiplicativo

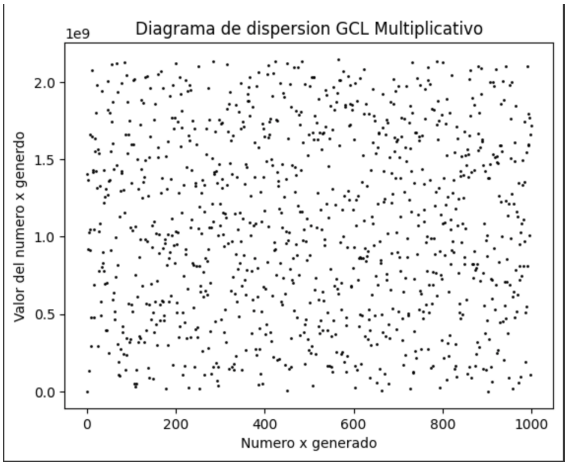

Figure 10: Diagrama de Dispersión GCL Multiplicativo con parámetros glibc

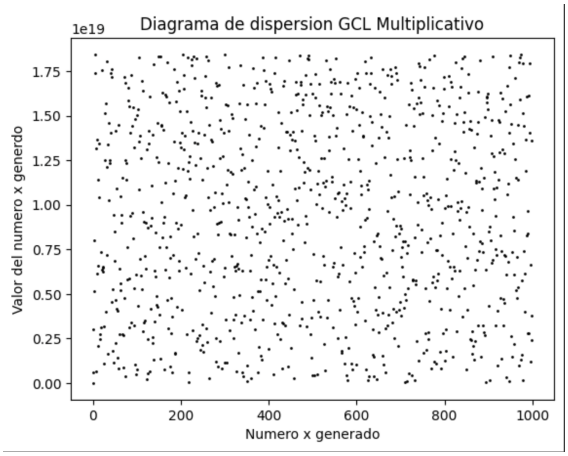

Figure 11: Diagrama de Dispersión GCL Multiplicativo con parámetros Knuth

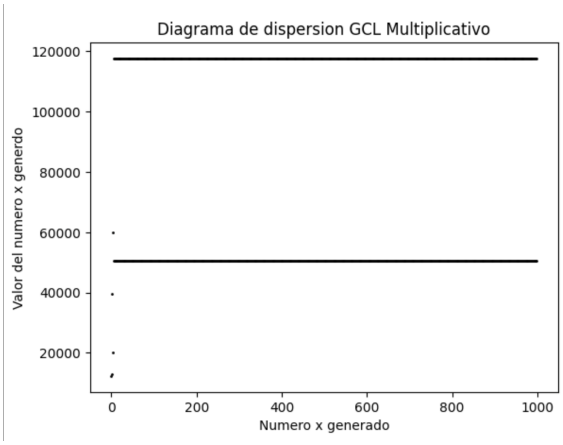

Figure 12: Diagrama de Dispersión GCL Multiplicativo con parámetros Java

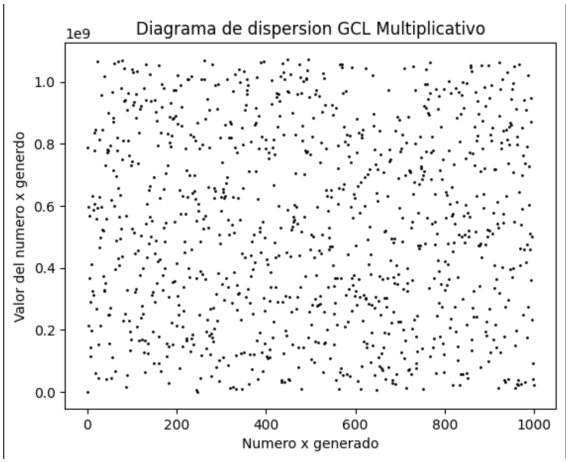

Figure 13: Diagrama de Dispersión GCL Multiplicativo con parámetros elaboración propia

Al igual que en el estudio del GCL Mixto, con los parámetros de glibc, Knuth y los nuestros, podemos considerar que es un buen generador, aprobando 3/4 tests. Sin embargo, los parámetros de Java tampoco generan buenos resultado en el Multiplicativo. De todas formas, podemos considerarlo un buen generador

4.4 Estudio del 'RandInt de Python'

En nuestros otros TPI, utilizamos el generador de números de Python para simular una ruleta. Sin embargo nunca habiamos revisado qué tan buen generador y qué tan confiable era. Estos fueron los resultados.

TEST	CHI CUADRADO	POKER	RUN ARRIBA Y ABAJO	RUN ARRIBA Y ABAJO DE LA MEDIA
	APROBADO	NO APROBADO	APROBADO	APROBADO

Figure 14: Tabla de Resultado de Python random.randInt

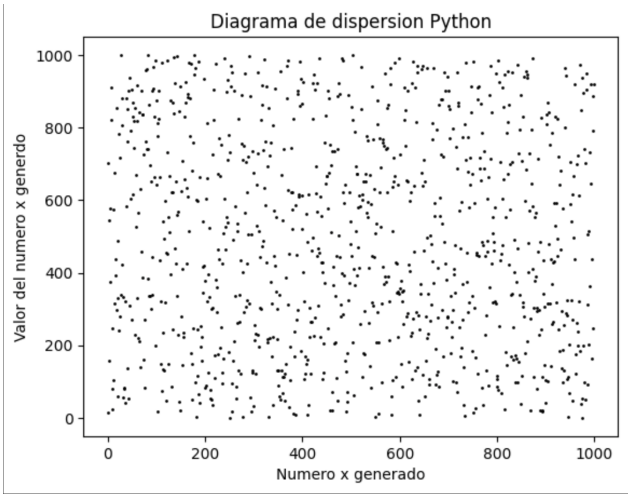

Figure 15: Diagrama de Dispersión Python

Podemos determinar que el generador de Python en confiable y podemos seguir utilizandolo para nuestrar simulaciones siguientes.

4.5 Generales.

Es un hecho que los Generadores Congruentes Lineales, ya sean Mixtos o Multiplicativos, pueden considerarse buenos generadores pseudoaleatorios, con los parámetros correctos. Además que el generador de Python es confiable para su uso. El generador de Cuadrados Medios ya ha sido descartado como generador pseudoaleatorio debido al patrón que se genera.

5 Bibliografía

Generación de Numeros Aleatorios. (s. f.). https://webs.um.es/mpulido/miwiki/lib/exe/fetch.php?media=wiki:simt1b.pdf

Ingeniería y Desarrollo. (s. f.). [Universidad del Norte Colombia]. https://www.redalyc.org/pdf/852/85215207001.pdf

(ruben.fcasal@udc.es), R. F. C., (rcao@udc.es), R. C., amp; (julian.costa@udc.es), J. C. (n.d.). Técnicas de simulación Y remuestreo. 2.1 Generadores congruenciales lineales. Retrieved April 25, 2023, from https://rubenfcasal.github.io/simbook/gen-cong.html Tests for Random Numbers. (n.d.). Retrieved April 25, 2023, from https://www.eg.bucknell.edu/ xmeng/Course/CS6337/Note/master/node42.html

Collegenote. (s/f). 2.Define and develop a Poker test for four-digitrandom numbers. A sequence of 10,000 random numbers, each of four digits hasbeen generated. The analysis of the numbers reveals that in 5120 numbers allfour digits are different, 4230 contain exactly one pair of like digits, 560contain two pairs, 75 have three digits of a kind and 15 contain all likedigits. Use Poker test to determine whether these numbers are independent.(Critical value of chi-square test for=0.05and N=4 is 9.49). Collegenote.net. Recuperado el 26 de abril de 2023, de https://www.collegenote.net/pastpapers/1244/question/

Mascagni, M. (s/f). Testing random numbers: Theory and practice. Fsu.edu. Recuperado el 26 de abril de 2023, de http://www.cs.fsu.edu/ mascagni/Testing.pdf

Okada, H., Umeno, K. (2017). Randomness evaluation with the discrete Fourier transform test based on exact analysis of the reference distribution. En arXiv [cs.CR]. http://arxiv.org/abs/1701.01960

Runs above and below the mean test. (s/f). Mathematics Stack Exchange. Recuperado el 26 de abril de 2023, de https://math.stackexchange.com/questions/85423/runs-above-and-below-the-mean-test

(S/f). Cartagena99.com. Recuperado el 26 de abril de 2023, de https://www.cartagena99.com/recursos/alumnos/apuntes/02_SIMULACION%20Numeros%20Aleatorios.pdf

https://www.collegenote.net/pastpapers/1244/question/

https://webs.um.es/mpulido/miwiki/lib/exe/fetch.php?media=wiki:simt1b.pdf

https://www.scribd.com/document/458615798/Prueba-Poker

https://www.statisticshowto.com/runs-test/

Kenton, W. (2007, mayo 30). Runs test: Definition, types, uses, and benefits. Investopedia. https://www.investopedia.com/terms/r/runs_test.asp

Patricio, H. (2021, diciembre 7). Generadores de números aleatorios y su importancia. The Dojo MX Blog. https://blog.thedojo.mx/2021/12/07/generadores-de-numeros-aleatorios-y-su-importancia-para-el-desarrollo.html