

# Taller # 01 | Generador de Números Pseudo Aleatorios y Pruebas de Bondad

Caicedo Hidalgo Geraldine [1527691] [geraldine.caicedo@correounivalle.edu.co]

**Abstracto** – Este documento presenta el análisis y explicación del taller número 1 de simulación computacional, el cual consiste en una implementación de un generador de números pseudo aleatorios GEM y comparar con el generador propio del lenguaje de programación usado empleando las pruebas de bondad, Chi Cuadrado, Poker 2 y 3.

**Índice de Términos** – pseudo aleatorios, generador, pruebas de bondad, GEM, chi cuadrado, poker.

## GENERACION DE NUMEROS PSEUDO ALEATORIOS

Para implementar el generador de números pseudo aleatorios se utilizó el lenguaje Java, utilizando el generador visto en clase (GEM) y el generador de números de aleatorios provisto por Java.

Se garantiza en el código que el generador tenga una semilla diferente de 0.

## PRUEBAS DE BONDAD

Se realizarán 3 pruebas diferentes a cada generador, Chi Cuadrado, Poker con dos decimales y con tres decimales.

Las pruebas se harán para dos cantidades de datos, 1.000 y 10.000.

La prueba de **Chi Cuadrado** se utiliza para comparar el comportamiento de los datos generados con la distribución uniforme  $U[0,1]$ .

En esta prueba se calcula el número de clases, los grados de libertad los cuales para las cantidades de datos es:

Para 1.000:

$$\begin{aligned}\text{numero de clases} &= \sqrt{\text{cantidad de datos}} \\ &= \sqrt{1.000} = 31.62 \cong 32\end{aligned}$$

$$\begin{aligned}\text{grados de libertad} &= \text{numero de clases} - 1 \\ &= 32 - 1 = 31\end{aligned}$$

Para 10.000:

$$\begin{aligned}\text{numero de clases} &= \sqrt{\text{cantidad de datos}} \\ &= \sqrt{10.000} = 100\end{aligned}$$

$$\begin{aligned}\text{grados de libertad} &= \text{numero de clases} - 1 \\ &= 100 - 1 = 99\end{aligned}$$

Se clasifican los  $n$  datos en  $\sqrt{n}$  clases de frecuencia obtenida, se calcula la frecuencia esperada y el chi cuadrado calculado para cada clase, como es uniforme, la frecuencia esperada es igual para todos.

$$FE = \text{numero de datos} / \text{numero de clases}$$

Que para las dos cantidades de datos sería:

$$n = 1.000; FE = 1.000 / 32 = 31.25$$

$$n = 10.000; FE = 10.000 / 100 = 100$$

Para calcular chi cuadrado calculado se utiliza la siguiente fórmula:

$$X_{cal}^2 = \sum (FE_i - FO_i)^2 / FE_i$$

Teniendo en cuenta que el valor de confianza dado para calcula chi cuadrado es  $\alpha = 0,1$  se tiene el valor del chi cuadrado critico según el número de datos.

$$n = 1.000; \quad X_{crit}^2 = 41.4217$$

$$n = 10.000; \quad X_{crit}^2 = 118.498$$

Las pruebas de **Poker** se utilizan para analizar si los datos generados se comportan como datos verdaderamente aleatorios.

Para esta prueba se toman  $k$  dígitos decimales de un número pseudo aleatorio como una mano de  $k$  cartas de un juego de cartas, que están numeradas de 0 a 9.

Si estos números son independientes, deben ocurrir los mismos patrones, y con las mismas probabilidades que en un juego de poker:

- Todas las cartas iguales
- Cartas en orden creciente o decreciente

Se buscan patrones que sucedan en secuencias aleatorias de números del 0 al 9, agrupados en sub secuencias de tamaño  $k$  (las manos). Los patrones deben ser mutuamente disjuntos y exhaustivos.

Se calculan las probabilidades  $p_i$  teóricas de cada patrón  $i$ .

Se aplica la prueba chi cuadrado donde las calases corresponde a los patrones, las frecuencias observadas se determinan con las manos y las frecuencias esperadas de un patrón  $i$  corresponde a  $p_i * n$

Para la prueba de **2** decimales las frecuencias observadas son:

- 2 números iguales
- 2 números diferentes

Para la prueba de **3** decimales las frecuencias observadas son:

- 3 números iguales
- 2 iguales y 1 diferente
- 3 números diferentes

## GENERADOR GEM

El generador GEM se implementa con la relación de recurrencia:

$$X_{n+1} = (a * X_n + c) \bmod m; \quad X_0 = \text{semilla}$$

$$R_n = X_n / m$$

Donde  $a$  es el multiplicador;  $c$  es el incremento y  $m$  es el modulo.

Se tiene  $0 \leq X_n < m$ , por eso  $0 \leq R_n < 1$ .

## Chi cuadrado

En la *Imagen 1* se muestra el resultado del test Chi Cuadrado para 1.000 números pseudo aleatorios con el generador GEM.

----- Chi Squared Test -----					
Classes Number: 32.0					
Degrees of Freedom: 31.0					
Rank	OF	EF			(EF-OF)^2/EF
[0.0 - 0.03125)	38	31.25	1.458		
[0.03125 - 0.0625)	27	31.25		0.578	
[0.0625 - 0.09375)	38	31.25		1.458	
[0.09375 - 0.125)	25	31.25		1.25	
[0.125 - 0.15625)	29	31.25		0.162	
[0.15625 - 0.1875)	28	31.25		0.338	
[0.1875 - 0.21875)	29	31.25		0.162	
[0.21875 - 0.25)	19	31.25		4.802	
[0.25 - 0.28125)	30	31.25		0.05	
[0.28125 - 0.3125)	29	31.25		0.162	
[0.3125 - 0.34375)	34	31.25		0.242	
[0.34375 - 0.375)	20	31.25		4.05	
[0.375 - 0.40625)	33	31.25		0.098	
[0.40625 - 0.4375)	43	31.25		4.418	
[0.4375 - 0.46875)	27	31.25		0.578	
[0.46875 - 0.5)	27	31.25	0.578		
[0.5 - 0.53125)	36	31.25	0.722		
[0.53125 - 0.5625)	26	31.25		0.882	
[0.5625 - 0.59375)	24	31.25		1.682	
[0.59375 - 0.625)	37	31.25		1.058	
[0.625 - 0.65625)	33	31.25		0.098	
[0.65625 - 0.6875)	31	31.25		0.002	
[0.6875 - 0.71875)	36	31.25		0.722	
[0.71875 - 0.75)	39	31.25		1.922	
[0.75 - 0.78125)	37	31.25		1.058	
[0.78125 - 0.8125)	34	31.25		0.242	
[0.8125 - 0.84375)	41	31.25		3.042	
[0.84375 - 0.875)	28	31.25		0.338	
[0.875 - 0.90625)	32	31.25		0.018	
[0.90625 - 0.9375)	19	31.25		4.802	
[0.9375 - 0.96875)	34	31.25		0.242	
[0.96875 - 1.0)	37	31.25	1.058		
Critical Chi Squared 41.4217					
Calculated Chi Squared: 38.272					
Total Obtained Frequency: 1000					
Past the Test?: true					

**Imagen 1.** Resultados Chi Cuadrado de 1.000 números con Generador GEM

En este caso se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

En la *Imagen 2* se muestra el resultado del test Chi Cuadrado para 10.000 números pseudo aleatorios con el generador GEM.

Chi Squared Test				
Classes Number: 100.0				
Degrees of Freedom: 99.0				
Rank	OF	EF	(EF-OF)^2/EF	
[0.0 - 0.01)	103	100.0	0.09	
[0.01 - 0.02)	96	100.0	0.16	
[0.02 - 0.03)	83	100.0	2.89	
[0.03 - 0.04)	114	100.0	1.96	
[0.04 - 0.05)	87	100.0	1.69	
[0.05 - 0.06000000000000005)	99	100.0	0.01	
[0.06000000000000005 - 0.07)	99	100.0	0.01	
[0.07 - 0.08)	90	100.0	1.0	
[0.08 - 0.09)	88	100.0	1.44	
[0.09 - 0.09999999999999999)	85	100.0	2.25	
[0.09999999999999999 - 0.10999999999999999)	113	100.0	1.69	
[0.10999999999999999 - 0.11999999999999999)	83	100.0	2.89	
[0.11999999999999999 - 0.12999999999999999)	111	100.0	1.21	
[0.12999999999999999 - 0.13999999999999999)	112	100.0	1.44	
[0.13999999999999999 - 0.15)	117	100.0	2.89	
[0.15 - 0.16)	100	100.0	0.0	
[0.16 - 0.17)	103	100.0	0.09	
[0.17 - 0.18000000000000002)	95	100.0	0.25	
[0.18000000000000002 - 0.19000000000000003)	100	100.0	0.0	
[0.19000000000000003 - 0.20000000000000004)	120	100.0	4.0	
[0.20000000000000004 - 0.21000000000000005)	92	100.0	0.64	
[0.21000000000000005 - 0.22000000000000006)	120	100.0	4.0	
[0.22000000000000006 - 0.23000000000000007)	101	100.0	0.01	
[0.23000000000000007 - 0.24000000000000007)	86	100.0	1.96	
[0.24000000000000007 - 0.25000000000000006)	105	100.0	0.25	
[0.25000000000000006 - 0.26000000000000006)	101	100.0	0.01	
[0.26000000000000006 - 0.27000000000000001)	103	100.0	0.09	
[0.27000000000000001 - 0.28000000000000001)	98	100.0	0.04	
[0.28000000000000001 - 0.29000000000000001)	110	100.0	1.0	
[0.29000000000000001 - 0.30000000000000001)	94	100.0	0.36	
[0.30000000000000001 - 0.31000000000000001)	93	100.0	0.49	
[0.31000000000000001 - 0.32000000000000001)	85	100.0	2.25	
[0.32000000000000001 - 0.33000000000000001)	100	100.0	0.0	
[0.33000000000000001 - 0.34000000000000001)	89	100.0	1.21	
[0.34000000000000001 - 0.35000000000000001)	91	100.0	0.81	
[0.35000000000000001 - 0.36000000000000001)	94	100.0	0.36	
[0.36000000000000001 - 0.37000000000000001)	91	100.0	0.81	
[0.37000000000000001 - 0.38000000000000001)	101	100.0	0.01	
[0.38000000000000001 - 0.39000000000000002)	112	100.0	1.44	
[0.39000000000000002 - 0.40000000000000002)	103	100.0	0.09	
[0.40000000000000002 - 0.41000000000000002)	88	100.0	1.44	
[0.41000000000000002 - 0.42000000000000002)	99	100.0	0.01	
[0.42000000000000002 - 0.43000000000000002)	97	100.0	0.09	
[0.43000000000000002 - 0.44000000000000002)	104	100.0	0.16	
[0.44000000000000002 - 0.45000000000000002)	95	100.0	0.25	
[0.45000000000000002 - 0.46000000000000002)	104	100.0	0.16	
[0.46000000000000002 - 0.47000000000000002)	111	100.0	1.21	
[0.47000000000000002 - 0.48000000000000002)	91	100.0	0.81	
[0.48000000000000002 - 0.49000000000000002)	108	100.0	0.64	
[0.49000000000000002 - 0.50000000000000002)	102	100.0	0.04	
[0.50000000000000002 - 0.51000000000000002)	107	100.0	0.49	
[0.51000000000000002 - 0.52000000000000002)	108	100.0	0.64	
[0.52000000000000002 - 0.53000000000000002)	103	100.0	0.09	
[0.53000000000000002 - 0.54000000000000003)	108	100.0	0.64	
[0.54000000000000003 - 0.55000000000000003)	108	100.0	0.64	
[0.55000000000000003 - 0.56000000000000003)	104	100.0	0.16	
[0.56000000000000003 - 0.57000000000000003)	95	100.0	0.25	
[0.57000000000000003 - 0.58000000000000003)	92	100.0	0.64	
[0.58000000000000003 - 0.59000000000000003)	107	100.0	0.49	
[0.59000000000000003 - 0.60000000000000003)	88	100.0	1.44	
[0.60000000000000003 - 0.61000000000000003)	102	100.0	0.04	
[0.61000000000000003 - 0.62000000000000003)	102	100.0	0.04	
[0.62000000000000003 - 0.63000000000000003)	116	100.0	2.56	
[0.63000000000000003 - 0.64000000000000003)	107	100.0	0.49	
[0.64000000000000003 - 0.65000000000000004)	85	100.0	2.25	
[0.65000000000000004 - 0.66000000000000004)	96	100.0	0.16	
[0.66000000000000004 - 0.67000000000000004)	107	100.0	0.49	
[0.67000000000000004 - 0.68000000000000004)	97	100.0	0.09	
[0.68000000000000004 - 0.69000000000000004)	109	100.0	0.81	
[0.69000000000000004 - 0.70000000000000004)	107	100.0	0.49	
[0.70000000000000004 - 0.71000000000000004)	116	100.0	2.56	
[0.71000000000000004 - 0.72000000000000004)	79	100.0	4.41	
[0.72000000000000004 - 0.73000000000000004)	97	100.0	0.09	
[0.73000000000000004 - 0.74000000000000004)	112	100.0	1.44	
[0.74000000000000004 - 0.75000000000000004)	84	100.0	2.56	
[0.75000000000000004 - 0.76000000000000005)	99	100.0	0.01	
[0.76000000000000005 - 0.77000000000000005)	112	100.0	1.44	
[0.77000000000000005 - 0.78000000000000005)	117	100.0	2.89	
[0.78000000000000005 - 0.79000000000000005)	79	100.0	4.41	
[0.79000000000000005 - 0.80000000000000005)	83	100.0	2.89	
[0.80000000000000005 - 0.81000000000000005)	107	100.0	0.49	
[0.81000000000000005 - 0.82000000000000005)	92	100.0	0.64	
[0.82000000000000005 - 0.83000000000000005)	96	100.0	0.16	
[0.83000000000000005 - 0.84000000000000005)	103	100.0	0.09	
[0.84000000000000005 - 0.85000000000000005)	100	100.0	0.0	
[0.85000000000000005 - 0.86000000000000005)	94	100.0	0.36	
[0.86000000000000005 - 0.87000000000000006)	91	100.0	0.81	
[0.87000000000000006 - 0.88000000000000006)	102	100.0	0.04	
[0.88000000000000006 - 0.89000000000000006)	94	100.0	0.36	
[0.89000000000000006 - 0.90000000000000006)	118	100.0	2.24	
[0.90000000000000006 - 0.91000000000000006)	98	100.0	0.04	
[0.91000000000000006 - 0.92000000000000006)	107	100.0	0.49	
[0.92000000000000006 - 0.93000000000000006)	90	100.0	1.0	
[0.93000000000000006 - 0.94000000000000006)	90	100.0	1.0	
[0.94000000000000006 - 0.95000000000000006)	93	100.0	0.49	
[0.95000000000000006 - 0.96000000000000006)	106	100.0	0.36	
[0.96000000000000006 - 0.97000000000000006)	103	100.0	0.09	
[0.97000000000000006 - 0.98000000000000006)	100	100.0	0.0	
[0.98000000000000006 - 0.99000000000000007)	111	100.0	1.21	
[0.99000000000000007 - 1.00000000000000007)	107	100.0	0.49	
Critical Chi Squared 118.498				
Calculated Chi Squared: 92.84				
Total Obtained Frequency: 10000				
Past the Test?: true				

Imagen 2. Resultados Chi Cuadrado de 10.000 números con Generador GEM

También aquí se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

## Poker 2

En la *Imagen 3* se muestra el resultado del test Poker con 2 decimales para 1.000 números pseudo aleatorios con el generador GEM.

Poker 2 Test			
Class	OF	EF	x2
2 Equals	102	100.0	0.04
2 Diferents	898	900.0	0.004444444444444444
-----			
Calculated Chi Squared:	0.04444444444444444		
Critical Chi Squared	2.7055		
Past the Test?: true			

Imagen 3. Resultados Poker 2 decimales de 1.000 números con Generador GEM

Aquí se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

En la *Imagen 4* se muestra el resultado del test Poker con 2 decimales para 10.000 números pseudo aleatorios con el generador GEM.

Poker 2 Test			
Class	OF	EF	x2
2 Equals	864	1000.0	18.496
2 Diferents	9136	9000.0	2.0551111111111111
-----			
Calculated Chi Squared:	20.551111111111111		
Critical Chi Squared	2.7055		
Past the Test?: false			

Imagen 4. Resultados Poker 2 decimales de 10.000 números con Generador GEM

En este caso se puede ver que el generador no paso la prueba, ya que el chi cuadrado critico es menor que el chi cuadrado calculado.

## Poker 3

En la *Imagen 5* se muestra el resultado del test Poker con 3 decimales para 1.000 números pseudo aleatorios con el generador GEM.

Poker 3 Test			
Class	OF	EF	x2
3 Equals	9	10.0	0.1
2 Equals, 1 Diferent	247	270.0	1.9592592592592593
3 Diferents	744	720.0	0.8
-----			
Calculated Chi Squared:	2.859259259259259		
Critical Chi Squared	4.6052		
Past the Test?: true			

Imagen 5. Resultados Poker 3 decimales de 1.000 números con Generador GEM

En esta prueba se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

En la *Imagen 6* se muestra el resultado del test Poker con 3 decimales para 10.000 números pseudo aleatorios con el generador GEM.

Poker 3 Test				
Class	OF	EF	x2	
3 Equals	95	100.0	0.25	
2 Equals, 1 Different	2700	2700.0	0.0	
3 Differents	7205	7200.0	0.0034722222222222	
Calculated Chi Squared: 0.2534722222222222				
Critical Chi Squared 4.6052				
Past the Test?: true				

**Imagen 6.** Resultados Poker 3 decimales de 10.000 números con Generador GEM

En este caso también se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

## GENERADOR JAVA

Al elegir el lenguaje de programación Java se utilizó como generador de números pseudo aleatorios alternativo `Util.Random` para la comparación con el generador GEM.

Este generador utiliza un generador lineal congruente y sus parámetros son tomados del generador UNIX `rand48` con una ligera variación en la función de la semilla.

El periodo de este generador es  $2^{48}$ , y no puede producir todas las combinaciones de pares de 2 enteros posibles, y sus bits no son equitativamente aleatorios.

### Chi Cuadrado

En la *Imagen 7* se muestra el resultado del test Chi Cuadrado para 1.000 números pseudo aleatorios con el generador Java.

Allí se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

En la *Imagen 9* se muestra el resultado del test Chi Cuadrado para 10.000 números pseudo aleatorios con el generador Java.

Chi Squared Test				
Classes Number: 32.0				
Degrees of Freedom: 31.0				
Rank	OF	EF	(EF-OF)	2/EF
[0.0 - 0.03125)	31	31.25	0.002	0.098
[0.03125 - 0.0625)	33	31.25	0.002	0.242
[0.0625 - 0.09375)	34	31.25	0.002	0.002
[0.09375 - 0.125)	31	31.25	0.002	0.018
[0.125 - 0.15625)	32	31.25	0.002	0.098
[0.15625 - 0.1875)	33	31.25	0.002	0.018
[0.1875 - 0.21875)	32	31.25	0.002	0.338
[0.21875 - 0.25)	28	31.25	0.338	0.338
[0.25 - 0.28125)	28	31.25	0.338	0.162
[0.28125 - 0.3125)	29	31.25	0.162	2.45
[0.3125 - 0.34375)	40	31.25	0.45	0.018
[0.34375 - 0.375)	35	31.25	0.45	0.098
[0.375 - 0.40625)	32	31.25	0.018	3.698
[0.40625 - 0.4375)	42	31.25	0.098	
[0.4375 - 0.46875)	33	31.25		
[0.46875 - 0.5)	35	31.25	0.45	0.002
[0.5 - 0.53125)	31	31.25		
[0.53125 - 0.5625)	14	31.25	9.522	1.682
[0.5625 - 0.59375)	24	31.25	0.578	0.578
[0.59375 - 0.625)	27	31.25	0.578	0.578
[0.625 - 0.65625)	27	31.25	0.578	0.578
[0.65625 - 0.6875)	27	31.25	0.578	0.002
[0.6875 - 0.71875)	37	31.25	0.018	0.018
[0.71875 - 0.75)	31	31.25	0.002	0.002
[0.75 - 0.78125)	32	31.25	0.018	0.002
[0.78125 - 0.8125)	31	31.25	0.002	0.338
[0.8125 - 0.84375)	28	31.25	0.018	0.018
[0.84375 - 0.875)	32	31.25	0.018	0.578
[0.875 - 0.90625)	32	31.25	0.578	1.458
[0.90625 - 0.9375)	27	31.25		
[0.9375 - 0.96875)	38	31.25		
[0.96875 - 1.0)	34	31.25	0.242	
Critical Chi Squared 41.4217				
Calculated Chi Squared: 25.151999999999997				
Total Obtained Frequency: 1000				
Past the Test?: true				

**Imagen 7.** Resultados Chi Cuadros de 1.000 números con Generador Java

En la *Imagen 9* se puede ver que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

### Poker 2

En la *Imagen 8* se muestra el resultado del test Poker con 2 decimales para 1.000 números pseudo aleatorios con el generador Java.

Poker 2 Test			
Class	OF	EF	x2
2 Equals	69	100.0	9.61
2 Differents	931	900.0	1.0677777777777777
Calculated Chi Squared: 10.677777777777777			
Critical Chi Squared 2.7055			
Past the Test?: false			

**Imagen 8.** Resultados Poker con 2 Decimales de 1.000 números con Generador Java

Este test muestra que el generador no paso la prueba, ya que el chi cuadrado critico es menor que el chi cuadrado calculado.

En la *Imagen 10* se muestra el resultado del test Poker con 2 decimales para 10.000 números pseudo aleatorios con el generador Java.

Chi Squared Test				
Classes Number: 100.0				
Degrees of Freedom: 99.0				
Rank	OF	EF	(EF-OF)²/EF	
[0.0 - 0.01]	115	100.0	2.25	
[0.01 - 0.02]	102	100.0	0.04	
[0.02 - 0.03]	102	100.0	0.04	
[0.03 - 0.04]	104	100.0	0.16	
[0.04 - 0.05]	95	100.0	0.25	
[0.05 - 0.060000000000000005]	111	100.0	1.21	
[0.06000000000000000005 - 0.07]	102	100.0	0.04	
[0.07 - 0.08]	95	100.0	0.25	
[0.08 - 0.09]	95	100.0	0.25	
[0.09 - 0.109999999999999999]	92	100.0		0.64
[0.109999999999999999 - 0.109999999999999999]	99	100.0		0.01
[0.109999999999999999 - 0.119999999999999999]	106	100.0		0.36
[0.119999999999999999 - 0.129999999999999999]	116	100.0		2.56
[0.129999999999999999 - 0.139999999999999999]	112	100.0		1.44
[0.139999999999999999 - 0.15]	102	100.0	0.04	
[0.15 - 0.16]	104	100.0	0.16	
[0.16 - 0.17]	102	100.0	0.04	
[0.17 - 0.180000000000000002]	97	100.0	0.09	
[0.18000000000000000002 - 0.190000000000000003]	87	100.0	1.69	
[0.19000000000000000003 - 0.200000000000000004]	97	100.0	0.09	
[0.20000000000000000004 - 0.210000000000000005]	96	100.0	0.16	
[0.21000000000000000005 - 0.220000000000000006]	111	100.0	1.21	
[0.22000000000000000006 - 0.230000000000000007]	79	100.0	4.41	
[0.23000000000000000007 - 0.240000000000000007]	119	100.0	3.61	
[0.24000000000000000007 - 0.250000000000000006]	90	100.0	1.0	
[0.25000000000000000006 - 0.260000000000000006]	97	100.0	0.09	
[0.26000000000000000006 - 0.270000000000000001]	115	100.0	2.25	
[0.27000000000000000001 - 0.280000000000000001]	110	100.0	1.0	
[0.28000000000000000001 - 0.290000000000000001]	104	100.0	0.16	
[0.29000000000000000001 - 0.300000000000000001]	75	100.0	6.25	
[0.30000000000000000001 - 0.310000000000000001]	80	100.0	4.0	
[0.31000000000000000001 - 0.320000000000000001]	85	100.0	2.25	
[0.32000000000000000001 - 0.330000000000000001]	94	100.0	0.36	
[0.33000000000000000001 - 0.3400000000000000014]	88	100.0	1.44	
[0.340000000000000000014 - 0.3500000000000000014]	113	100.0	1.69	
[0.350000000000000000014 - 0.3600000000000000015]	90	100.0	1.0	
[0.360000000000000000015 - 0.3700000000000000016]	104	100.0	0.16	
[0.370000000000000000016 - 0.3800000000000000017]	97	100.0	0.09	
[0.380000000000000000017 - 0.390000000000000002]	83	100.0	2.89	
[0.39000000000000000002 - 0.400000000000000002]	107	100.0	0.49	
[0.40000000000000000002 - 0.410000000000000002]	121	100.0	4.41	
[0.41000000000000000002 - 0.420000000000000002]	100	100.0	0.0	
[0.42000000000000000002 - 0.430000000000000002]	97	100.0	0.09	
[0.43000000000000000002 - 0.440000000000000002]	93	100.0	0.49	
[0.44000000000000000002 - 0.4500000000000000023]	103	100.0	0.09	
[0.450000000000000000023 - 0.4600000000000000024]	97	100.0	0.09	
[0.460000000000000000024 - 0.4700000000000000025]	108	100.0	0.04	
[0.470000000000000000025 - 0.4800000000000000026]	111	100.0	1.21	
[0.480000000000000000026 - 0.4900000000000000027]	107	100.0	0.49	
[0.490000000000000000027 - 0.500000000000000002]	107	100.0	0.49	
[0.50000000000000000002 - 0.510000000000000002]	95	100.0	0.25	
[0.51000000000000000002 - 0.520000000000000002]	97	100.0	0.09	
[0.52000000000000000002 - 0.530000000000000002]	103	100.0	0.09	
[0.53000000000000000002 - 0.540000000000000003]	118	100.0	3.24	
[0.54000000000000000003 - 0.550000000000000003]	94	100.0	0.36	
[0.55000000000000000003 - 0.560000000000000003]	102	100.0	0.04	
[0.56000000000000000003 - 0.570000000000000003]	111	100.0	1.21	
[0.57000000000000000003 - 0.580000000000000003]	94	100.0	0.36	
[0.58000000000000000003 - 0.590000000000000003]	86	100.0	1.96	
[0.59000000000000000003 - 0.600000000000000003]	89	100.0	1.21	
[0.60000000000000000003 - 0.610000000000000003]	90	100.0	0.0	
[0.61000000000000000003 - 0.620000000000000003]	99	100.0	0.01	
[0.62000000000000000003 - 0.630000000000000003]	99	100.0	0.01	
[0.63000000000000000003 - 0.640000000000000003]	91	100.0	0.81	
[0.64000000000000000003 - 0.650000000000000004]	108	100.0	0.64	
[0.65000000000000000004 - 0.660000000000000004]	97	100.0	0.09	
[0.66000000000000000004 - 0.670000000000000004]	109	100.0	0.81	
[0.67000000000000000004 - 0.680000000000000004]	102	100.0	0.04	
[0.68000000000000000004 - 0.690000000000000004]	101	100.0	0.01	
[0.69000000000000000004 - 0.700000000000000004]	99	100.0	0.01	
[0.70000000000000000004 - 0.710000000000000004]	105	100.0	0.25	
[0.71000000000000000004 - 0.720000000000000004]	104	100.0	0.16	
[0.72000000000000000004 - 0.730000000000000004]	101	100.0	0.01	
[0.73000000000000000004 - 0.740000000000000004]	105	100.0	0.25	
[0.74000000000000000004 - 0.750000000000000004]	102	100.0	0.04	
[0.75000000000000000004 - 0.760000000000000005]	87	100.0	1.69	
[0.76000000000000000005 - 0.770000000000000005]	84	100.0	2.56	
[0.77000000000000000005 - 0.780000000000000005]	114	100.0	1.96	
[0.78000000000000000005 - 0.790000000000000005]	110	100.0	1.0	
[0.79000000000000000005 - 0.800000000000000005]	112	100.0	1.44	
[0.80000000000000000005 - 0.810000000000000005]	106	100.0	0.36	
[0.81000000000000000005 - 0.820000000000000005]	86	100.0	1.96	
[0.82000000000000000005 - 0.830000000000000005]	112	100.0	1.44	
[0.83000000000000000005 - 0.840000000000000005]	99	100.0	0.01	
[0.84000000000000000005 - 0.850000000000000005]	89	100.0	1.21	
[0.85000000000000000005 - 0.860000000000000005]	89	100.0	1.21	
[0.86000000000000000005 - 0.870000000000000006]	110	100.0	1.0	
[0.87000000000000000006 - 0.880000000000000006]	114	100.0	1.96	
[0.88000000000000000006 - 0.890000000000000006]	115	100.0	2.25	
[0.89000000000000000006 - 0.900000000000000006]	114	100.0	1.96	
[0.90000000000000000006 - 0.910000000000000006]	103	100.0	0.09	
[0.91000000000000000006 - 0.920000000000000006]	108	100.0	0.64	
[0.92000000000000000006 - 0.930000000000000006]	94	100.0	0.36	
[0.93000000000000000006 - 0.940000000000000006]	90	100.0	1.0	
[0.94000000000000000006 - 0.950000000000000006]	81	100.0	3.61	
[0.95000000000000000006 - 0.960000000000000006]	95	100.0	0.25	
[0.96000000000000000006 - 0.970000000000000006]	93	100.0	0.49	
[0.97000000000000000006 - 0.980000000000000006]	97	100.0	0.09	
[0.98000000000000000006 - 0.990000000000000007]	89	100.0	1.21	
[0.99000000000000000007 - 1.000000000000000007]	92	100.0	0.64	
Critical Chi Squared 118.498				
Calculated Chi Squared: 97.26				
Total Obtained Frequency: 10000				
Past the Test?: true				

**Imagen 9.** Resultados Chi Cuadros de 10.000 números con Generador Java

```
----- Poker 2 Test -----
Class      OF      EF      x2      25.921
2 Equals   899      1000.0
2 Differents  9161     9000.0
2.88011111111111111113
-----
Calculated Chi Squared:  28.801111111111112
Critical Chi Squared:    2.7055
Past the Test?: false
```

**Imagen 10.** Resultados Poker con 2 Decimales de 10.000 números con Generador Java

Aquí se muestra que el generador no paso la prueba, ya que el chi cuadrado critico es menor que el chi cuadrado calculado.

## Poker 3

En la *Imagen 11* se muestra el resultado del test Poker con 3 decimales para 1.000 números pseudo aleatorios con el generador Java.

Poker 3 Test				
Class	OF	EF	x2	
3 Equals	7	10.0	0.9	
2 Equals, 1 Different	271	270.0	0.003703703703703704	
3 Differents	722	720.0	0.0055555555555555556	
Calculated Chi Squared:	0.9092592592592592			
Critical Chi Squared	4.6052			
Pass the Test?: true				

**Imagen 11.** Resultados Poker con 3 Decimales de 1.000 números con Generador Java

En esta imagen se muestra que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

En la *Imagen 12* se muestra el resultado del test Poker con 3 decimales para 10.000 números pseudo aleatorios con el generador Java.

```

----- Poker 3 Test -----
Class                OF                EF                x2
3 Equals              113                100.0              1.69
2 Equals, 1 Different 2732                2700.0              0.3792592592592592
3 Differents          7155                7200.0              0.28125
-----
Calculated Chi Squared: 2.350509259259259
Critical Chi Squared   4.6052
Past the Test?: true

```

**Imagen 12.** Resultados Poker con 3 Decimales de 10.000 números con Generador Java

Aquí se muestra que el generador paso la prueba, ya que el chi cuadrado critico es mayor que el chi cuadrado calculado.

## ANÁLISIS DE LOS RESULTADOS

Según los resultados arrojados por las pruebas, el generador GEM implementado solo falla para las pruebas **de Poker con dos decimales** con una cantidad de números generados de 10.000, y el generador propio del lenguaje de programación Java,

falla en las pruebas de independencia, **Poker con dos decimales** en las dos cantidades de números probadas, 1.000 y 1000.

Es decir que en las pruebas de uniformidad los dos generadores son aceptables, pero en las pruebas de independencia, pasan solo con la prueba de **Poker con tres decimales**.

## BIBLIOGRAFIA

- Diapositivas de Clase – Simulación Computacional [Marzo 2017]
- Randomness of bits with LCGs – Neil Coffey - UK - [2013]  
[[http://www.javamex.com/tutorials/random\\_numbers/lcg\\_bit\\_positions.shtml](http://www.javamex.com/tutorials/random_numbers/lcg_bit_positions.shtml)]
- `Java.lang.Random` falls “mainly in the planes” – Neil Coffey - UK - [2013]  
[[http://www.javamex.com/tutorials/random\\_numbers/lcg\\_planes.shtml](http://www.javamex.com/tutorials/random_numbers/lcg_planes.shtml)]
- How does `java.util.Random` work and how good is it – Neil Coffey - UK - [2013]  
[[http://www.javamex.com/tutorials/random\\_numbers/java\\_util\\_random\\_algorithm.shtml#.WMS6W39yyvB](http://www.javamex.com/tutorials/random_numbers/java_util_random_algorithm.shtml#.WMS6W39yyvB)]