

UNIFORMITY ASSESMENT IN PSEUDORANDOM NUMBERS GENERATED BY ARDUINO PLATFORMS

EVALUACIÓN DE UNIFORMIDAD EN NÚMEROS PSEUDOALEATORIOS GENERADOS POR PLATAFORMAS ARDUINO

PhD. Gabriel Elías Chanchí G.¹, PhD. Manuel Alejandro Ospina A.¹,
PhD. Wilmar Yesid Campo M.²

¹Universidad de Cartagena

Facultad de Ingeniería, Programa de Ingeniería de Sistemas
Campus Piedra de Bolívar. Cartagena, Bolívar, Colombia.
+57 (5) 6754782

E-mail: {gchanchig, mospinaa}@unicartagena.edu.co

²Universidad del Quindío

Facultad de Ingeniería, Programa de Ingeniería Electrónica
Carrera 15 Calle 12 Norte. Armenia, Quindío, Colombia.
+57 (6) 7359300

E-mail: wycampo@uniquindio.edu.co

Abstract: In specific contexts such as simulation models, it is necessary to have pseudo-random number generators (PNG) that have statistical uniformity as the main characteristic. Thus, in order to assess the relevance in the use of generators based on open hardware platforms as arduino, in this paper we conducted a study based on the uniformity in the distribution of pseudo-random numbers (PRN) generated through this type of boards. For the above, the statistical evaluation methods of the mean, variance, Kolmogorov-Smirnov, frequencies and distances were taken into account. For the development of this study, a software system was built consisting of a component for the generation of PRN on the arduino nano board and a component for the capture, visualization and analysis of the PRN generated, which was implemented using the Python Flask framework and the statistical functions provided by the Scipy library.

Keywords: Arduino, generator, pseudo-random, uniformity assessment.

Resumen: En contextos específicos como el de los modelos de simulación se requiere contar con generadores de números pseudoaleatorios (GNPA) que tengan como característica la uniformidad estadística. Así, con el fin de evaluar la pertinencia en el uso de generadores basados en plataformas de hardware libre como arduino, en este artículo se realizó un estudio sobre la uniformidad en la distribución de los números pseudoaleatorios (NPA) generados a través de este tipo de dispositivos. Se tuvieron en cuenta los métodos de evaluación estadística de la media, la varianza, Kolmogorov-Smirnov, las frecuencias y las distancias. Para el desarrollo del estudio, se construyó un sistema software conformado por un componente de generación de números pseudoaleatorios en un arduino nano y un componente para la captura, visualización y análisis de los números pseudoaleatorios generados, el cual fue implementado mediante la plataforma de trabajo “Flask” de “Python” y las funciones estadísticas provistas por la librería “Scipy”.

Palabras clave: Arduino, evaluación de uniformidad, generador, pseudoaleatorios.

1. INTRODUCCION

Uno de los aspectos relevantes en el estudio de sistemas complejos a través de modelos de simulación, es la generación de números pseudoaleatorios (NPA) (Coronel-Brizio *et al.*, 2007; Olmos y Acosta, 2006; García-Bosque 2019). Los NPA son generados mediante un algoritmo determinístico y un valor semilla que permite inicializarlo (Pérez *et al.*, 2011). Así, el algoritmo genera secuencias con base en operaciones iterativas que usan estados precedentes para la generación de nuevos valores (Rivas *et al.*, 2016).

A partir de lo anterior, las metodologías de simulación requieren de una fuente de generación de NPA que pueda garantizar una buena calidad estadística (Olmos y Acosta, 2006). Una de las formas en la que se evalúa la calidad de los NPA, es a través de la verificación de su propiedad de uniformidad, para lo cual se proponen diferentes métodos estadísticos de evaluación tales como: el de los promedios, el de la varianza, el de las distancias, entre otros (Coss, 2003). Gracias al surgimiento de diferentes plataformas de hardware libre, tales como Arduino, Raspberry PI, BeagleBoard entre otras, se ha abierto la posibilidad de usar dichas plataformas en diferentes contextos de aplicación (Quiroga *et al.*, 2017), siendo los generadores pseudoaleatorios uno de los posibles usos (Yu *et al.*, 2019; Fan *et al.*, 2016).

Por lo anterior, este artículo propone el desarrollo de un estudio sobre la uniformidad en la distribución de números pseudo-aleatorios generados mediante plataformas tipo arduino. Para la ejecución del estudio, se construyó un sistema software conformado por un componente de generación de números pseudoaleatorios en Arduino y un componente para la captura, visualización y análisis de los NPA, el cual fue implementado a través del “framework Flask” de Python (versión 1.1.1) y la librería de análisis estadístico “Scipy”. En lo referente al análisis, el sistema software hace uso de cinco métodos de evaluación de uniformidad: el de los promedios (media), la varianza, Kolmogorov-Smirnov, las frecuencias y el de la distancia.

El presente estudio pretende servir de referencia en cuanto a la consideración de las plataformas arduino, para el uso en diferentes contextos de aplicación que requieran la implementación de generadores pseudoaleatorios, como es el caso de los modelos de simulación. Del mismo modo, el sistema software propuesto, puede ser empleado en la evaluación de generadores aleatorios basados en otros tipos de dispositivos

(microprocesadores) derivados del hardware libre.

El artículo está organizado de la siguiente forma: en la sección 2 se presenta la metodología empleada para el desarrollo del presente estudio; en la sección 3 se describen los diferentes métodos de evaluación considerados; en la sección 4 se muestra el aplicativo web desarrollado para la evaluación de los diferentes métodos estadísticos presentados en la sección 3 sobre plataformas Arduino; en la sección 5 se describe un estudio de caso desarrollado a partir de los números generados por un Arduino nano; finalmente en la sección 6 se presentan las conclusiones y trabajos futuros derivados del presente trabajo.

2. METODOLOGÍA

Para el desarrollo metodológico se definieron cuatro fases: estudio de los métodos estadísticos de evaluación de uniformidad en números pseudoaleatorios (F_1), desarrollo del módulo de generación de números pseudoaleatorios, construcción del sistema software para la obtención, visualización y análisis de los números pseudoaleatorios (F_2) y estudio de caso (F_3) (ver Fig. 1).

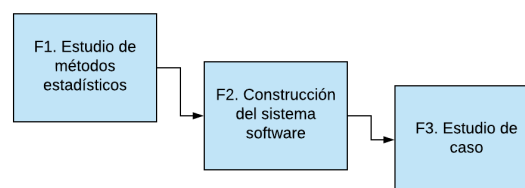


Fig. 1. Metodología propuesta

En la fase 1 (F_1) se procedió a identificar y seleccionar los métodos para la evaluación de la uniformidad en números pseudoaleatorios, obteniendo los siguientes: el método de los promedios, la varianza, Kolmogorov-Smirnov, las frecuencias y el de la distancia. En la fase 2 (F_2) se desarrolló un sistema software conformado por un generador de números pseudoaleatorios sobre la placa Arduino nano y una aplicación web para la obtención, visualización y análisis estadístico de los números proporcionados por el módulo de generación de números, haciendo uso del “framework Flask” de Python (versión 1.1.1) y de las librerías de análisis estadístico con las que cuenta este lenguaje. Finalmente, en la fase 4 (F_4) se desarrolló un caso de estudio con un conjunto de números pseudoaleatorios obtenidos a través del generador desarrollado en el Arduino nano.

3. MARCO CONCEPTUAL

En esta sección se describen los métodos de evaluación seleccionados para el desarrollo del

presente estudio. De esta manera, se presenta una descripción general del procedimiento para la evaluación de la uniformidad en números pseudoaleatorios el método de los promedios, el de la varianza, el de Kolmogorov-Smirnov, el de las frecuencias y el de las distancias (Coss, 2003; Montgomery y Runger, 2003; Montgomery, 2004).

3.1 Método de los Promedios

En el método de los promedios se calcula el valor de Z_o y si es menor que $Z_{\alpha/2}$ se considera el conjunto de NPA uniformemente distribuidos (ver Ecuación 1). El valor de $Z_{\alpha/2}$ es consultado en la tabla de distribución normal (Montgomery y Runger, 2003). N corresponde al total de números evaluados.

$$Z_o = \frac{(\text{media} - \frac{1}{2})\sqrt{N}}{\sqrt{1/12}} \quad (1)$$

3.2 Método de la Varianza

En el método de las varianzas se calcula la varianza del conjunto de números y se determina si está comprendida entre los límites inferior (LI) y superior (LS), los cuales son calculados teniendo en cuenta la tabla de distribución de chi cuadrado (Montgomery, 2004) para un valor de confianza determinada (α) y (n-1) grados de libertad (ver Ecuaciones 2 y 3).

$$LI = \frac{X_{1-\alpha}^2}{12(n-1)} \quad (2)$$

$$LS = \frac{X_{\alpha}^2}{12(n-1)} \quad (3)$$

3.3 Método de Kolmogorov-Smirnov

En el método de Kolmogorov-Smirnov (KS) se obtienen el valor máximo del estadístico de contraste D_n y se verifica si este valor es menor que el valor crítico obtenido a partir de la tabla de KS (Montgomery y Runger, 2003), con un valor de confianza α determinado. $F_n(x)$ es la distribución empírica y $F_o(x)$ la distribución hipotética (ver Ecuación 4).

$$D_n = \max (F_n(x) - F_o(x)) \quad (4)$$

3.4 Método de las Frecuencias

Dentro del método de las frecuencias se obtiene el valor de X_o^2 y se verifica si dicho valor es menor que el valor de $X_{\alpha, n-1}^2$, el cual es obtenido a partir de la tabla de distribución de chi cuadrado (Montgomery y Runger, 2003) para un α determinado y para n-1 grados de libertad, siendo

n la cantidad de intervalos definidos, F_{oi} es la frecuencia observada y F_{Ei} es la frecuencia esperada (ver Ecuación 5).

$$X_o^2 = \sum_{i=1}^m \frac{(F_{oi} - F_{Ei})^2}{F_{Ei}} \quad (5)$$

3.5 Método de las Distancias

En el método de las distancias o métodos de los huecos se obtiene el valor de X_o^2 para un intervalo (a,b) determinado y se verifica si dicho valor es menor que el valor de $X_{\alpha, n}^2$, el cual es obtenido a partir de la tabla de distribución de chi cuadrado para un nivel de confiabilidad α determinado y para n grados de libertad (Montgomery & Runger, 2003), siendo n la cantidad de huecos considerados, F_{oi} es la frecuencia observada y F_{Ei} es la frecuencia esperada (ver Ecuación 6).

$$X_o^2 = \sum_{i=1}^m \frac{(F_{oi} - F_{Ei})^2}{F_{Ei}} \quad (6)$$

Para la obtención de la frecuencia esperada (F_{Ei}) es necesario calcular la probabilidad del tamaño del hueco, la cual se determina mediante las Ecuaciones 7 y 8 (Montgomery y Runger, 2003). Así P_i es la probabilidad para un determinado tamaño y P_n la probabilidad para el número total de huecos considerados.

$$P_i = \theta(1 - \theta)^i \quad (7)$$

$$P_n = (1 - \theta)^n \quad (8)$$

4. SISTEMA SOFTWARE

En esta sección se presenta lo referente al diseño y la implementación del sistema software empleado en la evaluación de la uniformidad de números pseudoaleatorios generados por plataformas Arduino.

3.1 Diseño del Sistema Software

En la Fig. 2 se presenta el diagrama de bloques del sistema software desarrollado, en el cual se muestran los cuatro módulos funcionales: generación, captura, visualización y análisis.

Como se mencionó anteriormente, el módulo de generación fue implementado dentro de la tarjeta Arduino nano, mientras que los módulos de captura, visualización y análisis operan desde una aplicación web desarrollada en el “framework Flask” de Python. En el módulo de generación desde una tarjeta Arduino se crean números pseudoaleatorios cada 100 milisegundos, los cuales son enviados a través del puerto serial. En

el módulo captura se establece desde la aplicación web la conexión con el puerto serial y se obtienen de manera periódica los números enviados desde la tarjeta Arduino, mientras los números no se repitan, si los números se repiten, se detiene la captura desde el Arduino. En cuanto cada número va siendo capturado se va normalizando en el intervalo de 0 a 1 dividiendo entre 1000. Dentro del módulo de visualización, a medida que los números pseudoaleatorios son recibidos a través del puerto serial, se van presentando y graficando en tiempo real dentro de un gráfico en la aplicación web. Finalmente, cuando la captura de los datos desde el Arduino ha terminado, en el módulo de análisis se realiza la evaluación de los números capturados haciendo uso de los métodos estadísticos seleccionados y presentados en la sección 3.

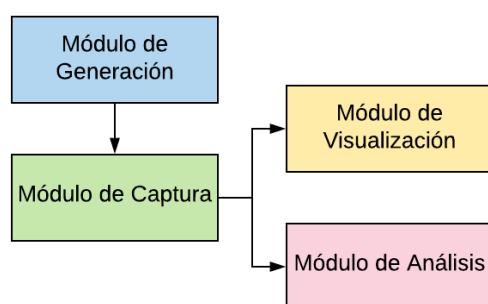


Fig. 2. Diagrama de Bloques

Un diagrama que ilustra el proceso seguido por el sistema software para la generación, captura, visualización y análisis de los números pseudoaleatorios se presenta en la Fig. 3.

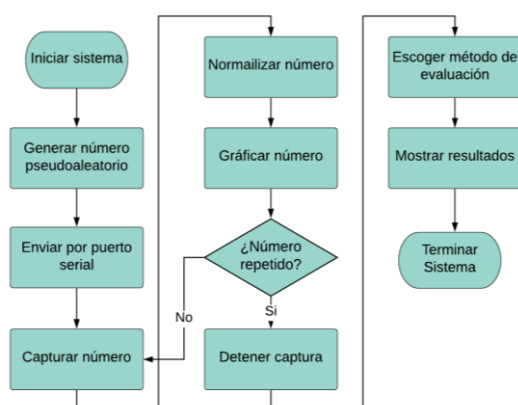


Fig. 3. Diagrama de Flujo del Sistema Software

Cuando el sistema da inicio, se generan números pseudoaleatorios desde el Arduino entre 1 y 1000, y son enviados por el puerto serial. Desde la aplicación web generada en “Flask” con Python se captura periódicamente desde el puerto serial los números generados por el Arduino y se procede con la normalización de los mismos entre 0 y 1. A medida que los números pseudoaleatorios son capturados se visualizan en tiempo real mediante

una gráfica en el aplicativo web. En el momento que se detecte en el módulo de captura del aplicativo web que se repite un número se detiene automáticamente la captura y se muestran los números capturados. Finalmente, el usuario puede escoger el método de evaluación estadística a aplicar de los cinco considerados en este trabajo. El aplicativo web mediante el uso de los métodos estadísticos proporcionados por Python realiza la evaluación y presenta los resultados indicando si el conjunto de números pseudoaleatorios generados pasan o no pasan la prueba de uniformidad y por ende de aleatoriedad.

3.1 Implementación del sistema software

A nivel de implementación, en la Fig. 4 se muestra un diagrama con las tecnologías y/o librerías empleadas para la implementación del sistema software propuesto en este artículo.

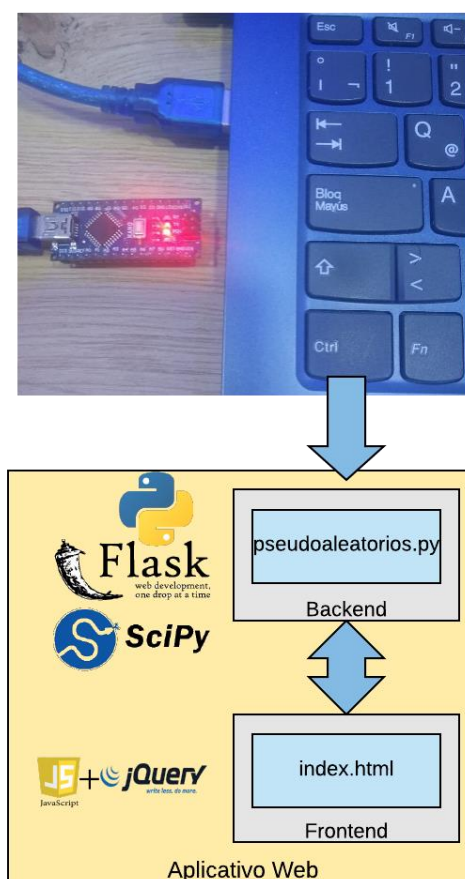
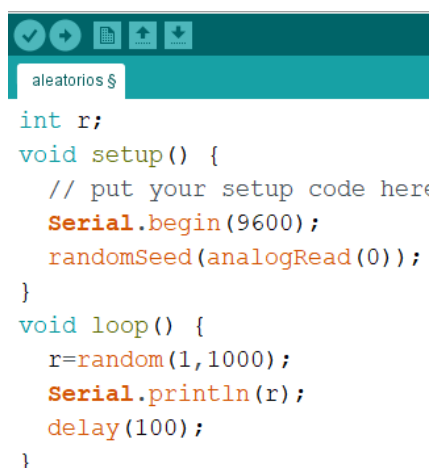


Fig. 4. Diagrama de Implementación

En lo referente al componente de generación de números pseudoaleatorios se hizo uso del Arduino nano (ver Fig. 4), en la cual partiendo de una semilla obtenida mediante la lectura del voltaje percibido en una entrada analógica (A0), se generan números pseudoaleatorio entre 1 y 1000 (ver Fig. 5), los cuales son enviados a través del puerto serial al componente de captura, visualización y análisis.



```

aleatorios $

int r;
void setup() {
  // put your setup code here
  Serial.begin(9600);
  randomSeed(analogRead(0));
}
void loop() {
  r=random(1,1000);
  Serial.println(r);
  delay(100);
}

```

Fig. 5. Generador de números pseudoaleatorios

El componente de captura, visualización y análisis se implementó a través de una aplicación web en Python usando el “framework Flask”. A nivel del “backend” de la aplicación web, se hizo uso de la librería “pyserial” de Python para la captura de los datos enviados por el generador de números aleatorios desde el puerto serial. Del mismo modo, el análisis estadístico de los datos capturados se realizó con apoyo de la librería “scipy” de Python, la cual permite realizar la consulta de las tablas de distribución normal, chi cuadrado y de Kolmogorov-Smirnov. Por su parte en el “frontend” de la aplicación web es posible visualizar la gráfica de los números generados en tiempo real gracias a la librería “Canvas.JS” y a la librería “JQuery” de “Javascript”. Del mismo modo en el “frontend” de la aplicación web se muestra el conjunto de números generados y el resultado de la aplicación de uno de los métodos considerados al listado de números (ver Fig. 6 y 7).

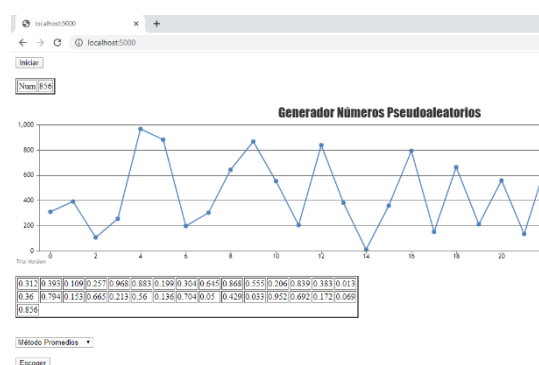


Fig. 6. Diagrama de Implementación

En la Fig. 6 se presenta la interfaz del aplicativo web y por ende del sistema desarrollado. En esta interfaz al presionar el botón iniciar, se da comienzo a la captura, normalización y visualización mediante una gráfica de los números hasta que el generador repita un número. Cuando la captura termina se presenta el listado de

números obtenidos a partir del generador y se habilitan las opciones de análisis de uniformidad de los números.

Una vez el usuario escoge uno de los cinco posibles métodos de evaluación, la interfaz gráfica presenta los resultados de la evaluación. En la Fig. 7 se presentan los resultados de aplicar el método de la varianza al conjunto de números pseudoaleatorios generados por la tarjeta Arduino nano.

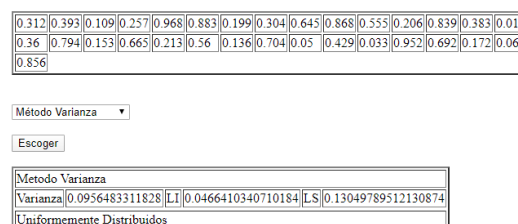


Fig. 7. Evaluación de uniformidad

5. ESTUDIO DE CASO

Se realizó mediante el sistema propuesto un estudio de la uniformidad con los números generados por la plataforma Arduino nano. Como criterio, se tomó en cuenta una muestra de números superior a 50. Así, en la Fig. 8 se muestra una captura de pantalla de los 68 números pseudoaleatorios obtenidos mediante el Arduino nano. Del mismo modo, se precisa que para cada una de las pruebas realizadas, se tomó en consideración un 95% de confianza ($\alpha = 0.05$).

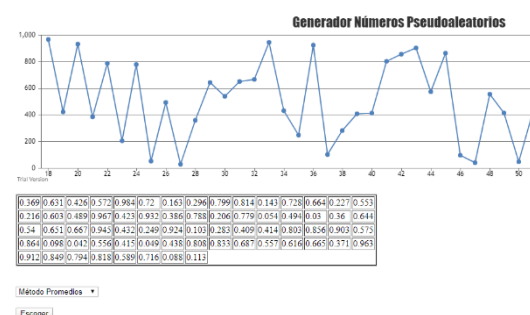


Fig. 8. Datos capturados

A partir de los resultados presentados en la Fig. 8, en la Tabla 1 se presentan los números a los cuales se les aplicó la prueba de uniformidad en el presente estudio.

Una vez aplicado el método de los promedios a los números de la Tabla 1, en la Tabla 2 se presentan los resultados de dicha prueba. Teniendo en cuenta que el valor de Z_0 es menor que $Z_{\alpha/2}$ se concluye que los números se encuentran uniformemente distribuidos. Cabe mencionar que, para el cálculo de LI, LS, y $Z_{\alpha/2}$ se consideró la tabla de distribución normal para el valor de $\alpha/2 = 0.025$.

Tabla 1: Tabla de números pseudoaleatorios

Id	Num	Id	Num	Id	Num
1	0.369	24	0.206	47	0.098
2	0.631	25	0.779	48	0.042
3	0.426	26	0.054	49	0.556
4	0.572	27	0.494	50	0.415
5	0.984	28	0.03	51	0.049
6	0.72	29	0.36	52	0.438
7	0.163	30	0.644	53	0.808
8	0.296	31	0.54	54	0.833
9	0.799	32	0.651	55	0.687
10	0.814	33	0.667	56	0.557
11	0.143	34	0.945	57	0.616
12	0.728	35	0.432	58	0.665
13	0.664	36	0.249	59	0.371
14	0.227	37	0.924	60	0.963
15	0.553	38	0.103	61	0.912
16	0.216	39	0.283	62	0.849
17	0.603	40	0.409	63	0.794
18	0.489	41	0.414	64	0.818
19	0.967	42	0.803	65	0.589
20	0.423	43	0.856	66	0.716
21	0.932	44	0.903	67	0.088
22	0.386	45	0.575	68	0.113
23	0.788	46	0.864		

Tabla 2: Tabla métodos de los promedios

Variable	Valor
Promedio	0.545
LI	0.4314
LS	0.5686
Zo	1.2833
Z $\alpha/2$	1.96

Al aplicar el método de la varianza al conjunto de números pseudoaleatorios de la Tabla 1, se obtienen los resultados presentados en la Tabla 3. Considerando que el valor de la varianza se encuentra entre el límite inferior (LI) y el límite superior (LS), los números pseudoaleatorios se encuentran uniformemente distribuidos. Para el cálculo de LI y LS se hizo uso de la tabla de la distribución chi cuadrado con $\alpha/2 = 0.025$ y $n-1=67$ grados de libertad.

Tabla 3: Tabla método de la varianza

Variable	Valor
Varianza	0.0781
LI	0.5754
LS	0.1138

Cuando se aplica el método de Kolmogorov-Smirnov a los números de la Tabla 1, se obtienen los resultados presentados en la Tabla 4. Teniendo en cuenta que el valor máximo de los dos valores estadísticos de Kolmogorov-Smirnov (D1) es menor que el valor crítico, se concluye que los números pseudoaleatorios se encuentran uniformemente distribuidos. Para el cálculo del valor crítico se hizo uso de un $\alpha=0.05$ para un tamaño de la muestra de 68.

Tabla 4: Tabla método de Kolmogorov-Smirnov

Variable	Valor
D1	0.1247
D2	0.018
V. Crítico	0.1620

De la misma forma, al aplicar el método de las frecuencias a los números pseudoaleatorios de la Tabla 1, se obtienen los resultados de la Tabla 5. Considerando que el valor de X_0^2 es menor que el valor de $X_{\alpha,n-1}^2$, se concluye que los números pseudoaleatorios evaluados están uniformemente distribuidos. Para el cálculo de $X_{\alpha,n-1}^2$ se hizo uso de $\alpha=0.05$ con $n-1=67$ grados de libertad. Del mismo modo se consideraron 5 subintervalos para la clasificación de los números.

Tabla 5: Tabla método de las Frecuencias

Variable	Valor
F _{Ei}	13.6
X_0^2	3.1765
$X_{\alpha,n-1}^2$	9.4877

Finalmente, al aplicar el método de las distancias sobre los números de la Tabla 1, se obtienen los resultados presentados en la Tabla 6. Teniendo en cuenta que el valor de X_0^2 es mayor que el valor de $X_{\alpha,n}^2$, se concluye que los números pseudoaleatorios evaluados no están uniformemente distribuidos según este método. Para el cálculo de $X_{\alpha,n}^2$ se consideró $\alpha=0.05$ y como grados de libertad $n=3$. Del mismo modo se consideraron los valores de $\alpha=0.3$ y $\beta=0.7$.

Tabla 6: Tabla método de las distancias

i	Pi	FOi	FEi
0	0.4	11.6	14
1	0.24	6.96	7
2	0.144	4.17	1
3	0.216	6.26	7
Total	1	29.0	29.0
	X_0^2		10.576
	$X_{\alpha,n}^2$		7.814

6. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se realizó un estudio sobre la uniformidad de los números generados mediante una Arduino nano. Mediante este estudio se evaluaron 5 métodos estadísticos con el fin de determinar la pertinencia de las plataformas basadas en Arduino como opción para la construcción de generadores pseudoaleatorios que puedan aplicarse en áreas como la simulación.

Como aporte adicional el presente trabajo propuso un sistema software para la evaluación de la uniformidad estadística de números pseudoaleatorios generados mediante la Arduino nano, el cual puede ser empleado para evaluar la uniformidad en otro tipo de plataformas derivadas del hardware libre.

Como resultado del estudio de caso realizado en el presente artículo, se obtuvo que el generador de números pseudoaleatorios basado en la placa arduino pasó 4 de los 5 métodos de evaluación estadística considerados, por lo que dentro del caso de estudio tuvo una efectividad en las pruebas del 80%. Lo anterior lo convierte en una buena opción de cara al uso dentro de diferentes contextos de aplicación como es el caso de los modelos de simulación.

Finalmente, como trabajo futuro, se pretende extender el estudio realizado en este trabajo a otras plataformas de hardware libre, con el fin de establecer un comparativo con los resultados presentados en este artículo, además es posible hacer una extensión de la evaluación de los métodos propuestos a variables aleatorias, así como evaluar las propiedades no solo de uniformidad si no de independencia en este tipo de plataformas.

REFERENCIAS

- Coronel-Brizio, H., Hernández-Montoya, A., y Jimenez-Montaña, M. (2007). Una prueba empírica de generadores de numeros pseudoaleatorios mediante un proceso de decaimiento exponencial. *Revista Mexicana de Física*, 53(5), 350-357.
- Coss, R. (2003). *Simulación, Un enfoque práctico*. México: Editorial Limusa.
- Fan, H., Chen, M., y Gao, S. (2016). "Corrected runs distribution test for pseudorandom number generators". *Electronics Letters*, vol. 52, no. 4, pp. 281-283.
- García-Bosque, M., Pérez-Resca, C., Sánchez-Azqueta, C., Aldea y Celma, S. (2019). "Chaos-Based Bitwise Dynamical Pseudorandom Number Generator On FPGA". *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 1, pp. 291-293.
- Montgomery, D. (2004). *Diseño y análisis de experimentos*. México, D.F.: Limusa.
- Montgomery, D., & Runger, G. (2003). *Probabilidad y estadística aplicada a la ingeniería*. México, D.F.: Limusa.
- Olmos, A., & Acosta, J. (2006). *Generación de números pseudoaleatorios eficientes en microcomputadoras*. México: Universidad Nacional Autónoma de México .
- Pérez, C., Mariño, S., & López, M. (2011). Desarrollo de generadores de números pseudoaleatorios en Octave. *TE & ET: Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*(6), 24-29.
- Quiroga, E., Jaramillo, S., Campo, W., & Chanchí, G. (2017). Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT. *Revista Ibérica de Sistemas e Tecnologías de Informação* (24), 39-56.
- Rivas, K., Lupera, P., & Tapintuña, C. (2016). *Implementación de un generador de números pseudoaleatorios sobre un FPGA para aplicaciones de dispersión de energía en transmisiones de televisión y radio digital*. Quito-Ecuador: Escuela Politécnica Nacional.
- Yu, F., Li, L., He, B., Liu, L., Qian, S. (2019). "Design and FPGA Implementation of a Pseudorandom Number Generator Based on a Four-Wing Memristive Hyperchaotic System and Bernoulli Map". *IEEE Access*, vol. 7, pp. 181884-181898.