

# Tarea III: Evaluación de Aleatoriedad Uniforme

Profesor: Eddy Ramírez

## 1 Objetivo

El objetivo de esta tarea es implementar y ejecutar diversas pruebas estadísticas para evaluar la aleatoriedad de secuencias numéricas generadas en diferentes lenguajes de programación. Cada estudiante debe programar las pruebas y preparar una presentación de los resultados.

## 2 Pruebas Estadísticas a Implementar

Usted debe de programar en el lenguaje de su preferencia (C, C++, Java, Python, Erlang, Scheme) cada una de las pruebas estadísticas citadas en clase (que se muestran a continuación).

1. **Prueba de Promedio:** Verificar si el promedio de los números generados está en el rango esperado.<sup>1</sup>
2. **Prueba de Varianza:** Evaluar si la varianza de los números generados coincide con la varianza teórica.<sup>2</sup>
3. **Prueba de Corridas:** Analizar la secuencia de números para determinar la cantidad de corridas (secuencias de números similares).<sup>3</sup>
4. **Prueba de Huecos con Dígitos:** Contar los huecos en la secuencia de números y verificar la distribución de dígitos.<sup>4</sup>
5. **Prueba de Huecos con Números:** Similar a la anterior, pero se basa en la distribución de números en la secuencia.<sup>5</sup>

---

<sup>1</sup>Considerar que se debe calcular si no se trata de una distribución  $[0,1[$

<sup>2</sup>ídem

<sup>3</sup>ídem

<sup>4</sup>ídem

<sup>5</sup>ídem

6. **Prueba de Póker:** Evaluar la frecuencia de combinaciones específicas en la secuencia.<sup>6</sup>
7. **Prueba de Series:** Verificar la cantidad de series en la secuencia de números.<sup>7</sup>

## 3 Lenguajes de Programación a mostrar

Utilizando cada uno de los lenguajes de programación que se mencionan a continuación, debe generar muestras de números según se indica

- **Java** ( $[0,1[$ )
- **Erlang** ( $[0,1[$ )
- **Python** ( $[0,1[$  y  $\{1,2,3,4,5,6\}$ )
- **C** ( $\{1,2,3,4\}$  y  $\{1,2,3,4,5,6,7,8\}$ )
- **Scheme/Racket** (números enteros en el rango  $[1, 20]$ )
- **Otro Lenguaje:** El estudiante debe elegir un lenguaje adicional de su preferencia (por ejemplo, JavaScript, Ruby, etc.) y un intervalo cómodo acorde con el lenguaje seleccionado.

## 4 Requisitos de Implementación

- Generar 1 millón de números aleatorios para cada lenguaje.
- Realizar cada prueba sobre las mismas secuencias generadas de cada lenguaje y almacenar los resultados y tablas estadísticas.

---

<sup>6</sup>ídem

<sup>7</sup>ídem

- Determinar si cada muestra de cada lenguaje pasa o no cada una de las pruebas estadísticas.

## 5 Presentación

- Crear una presentación utilizando **Beamer** en LaTeX.
- La presentación debe tener una duración aproximada de **10 minutos**.
- Incluir los siguientes elementos en la presentación:
  - Introducción a las pruebas estadísticas.
  - Datos teóricos de cada muestra (fórmulas).
  - Resultados de cada prueba para cada muestra de cada lenguaje.
  - Conclusiones sobre la aleatoriedad de los números generados.
  - Gráficos y tablas que muestren los resultados.

## 6 Consideraciones Finales

- Documentar código adecuadamente.
- Realiza pruebas exhaustivas para asegurar la validez de los resultados.
- Estar preparados para una ronda de preguntas al finalizar su presentación