

---

# TP 2.1 - GENERADORES PSEUDOALEATORIOS

---

**Juan I. Torres**

Universidad Tecnológica Nacional - FRRO  
Zeballos 1341, S2000, Argentina  
orkuan@gmail.com

29 de abril, 2025

## ABSTRACT

El siguiente documento tiene por objetivo detallar el trabajo de clase que debe realizarse para introducirnos en uno de los elementos fundamentales para gran parte de las simulaciones, esto son los **generadores de números pseudoaleatorios**.

**Keywords** Simulación · generadores pseudoaleatorios

## 1 Introducción

Un número pseudoaleatorio es un número generado en un proceso que parece producir números al azar, pero no lo hace realmente. Las secuencias de números pseudoaleatorios no muestran ningún patrón o regularidad aparente desde un punto de vista estadístico, a pesar de haber sido generadas por un algoritmo completamente determinista, ante las mismas condiciones iniciales producen siempre el mismo resultado.

## 2 Descripción del trabajo

El trabajo de investigación consiste en construir programas en lenguaje Python 3.x que generen números pseudoaleatorios y que estos se comporten como se espera. Para esto se debe tener en cuenta lo siguientes temas a investigar:

- Generadores de números aleatorios reales.
- Generadores de números pseudoaleatorios (Método de los cuadrados, GCLs, otros).
- Test para determinar el comportamiento de los generadores.

Se pide programar al menos dos generadores de números pseudoaleatorios en particular el generador GCL del cual se debe testear con al menos cuatro pruebas para determinar la calidad de generación. También se pide comparar los generadores programados con otros (incluyendo el que posee el lenguaje Python).

A modo de introducción y ejemplos de código se deja un link en la sección de **Recursos online obligatorios**.

### 2.1 Exposición de los resultados y análisis de los mismos

Los resultados se deben analizar y luego concluir sobre su comportamiento obtenido y esperado.

Se aconseja el uso de tablas para mostrar los datos de los distintos generadores y pruebas. En otros casos puede ser más convenientes el uso de imágenes. Ver los siguientes ejemplos:

### 2.2 Presentación del trabajo y entrega

Este trabajo de investigación y al igual que los anteriores se debe presentar obligatoriamente en formato  $\text{\LaTeX}$ .

El contenido mínimo a entregar es:

Generador	Tipo de Generador	Test X	Test Y	...
GCL	Pseudoaleatorio	0.4	OK	...
ERNIE	Aleatorio	0.2	ERROR	...
...	...	...	...	...

Tabla 1: Ejemplo de exposición de resultados

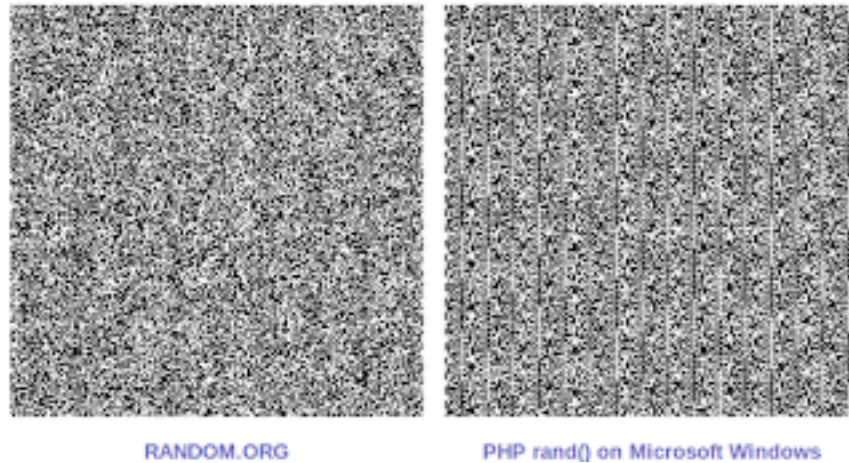


Figura 1: Comparación entre valores de dos generadores distintos. Imagen tomada de <http://www.random.org/analysis/>.

- Código completo en Python 3.x. con los generadores y tests.
- Informe en formato Latex con introducción, gráficas, fórmulas empleadas, conclusiones, referencias (hay un apartado para esto) y cualquier otra información que se quiera agregar.

### 3 Recursos online obligatorios

Volveremos a usar el template  $\LaTeX$  de la Cornell University por su sencillez:

<https://es.overleaf.com/latex/templates/style-and-template-for-preprints-arxiv-bio-arxiv/pkzcrhzcxdxmc>

Un resumen que va a ayudar a encontrar mucha otra información:

<https://tereom.github.io/est-computacional-2018/numeros-pseudoaleatorios.html>

Acá encontrarán información de varios tipos tests que pueden realizarse:

<https://www.random.org/analysis/>

Se recomienda leer con detenimiento el siguiente informe:

<https://www.random.org/analysis/Analysis2005.pdf>

### 4 Recursos online sugeridos

<https://www.youtube.com/watch?v=FtEe0IlK6Hc>