

## Propuestas de Proyectos de Investigación – Procesos Estocásticos

Este documento presenta una serie de propuestas de proyectos de investigación orientadas a la aplicación de los procesos estocásticos en diferentes ámbitos de la informática. Los temas integran aplicaciones reales en monitorización, redes, seguridad informática, bases de datos y sistemas distribuidos, junto con otros enfoques clásicos de modelado estocástico. Cada tema está diseñado para fomentar la integración de la teoría probabilística, el análisis matemático, la simulación computacional y la interpretación de fenómenos aleatorios mediante herramientas tecnológicas como Python y R (Opcional).

| Nº | Tema del Proyecto   | Objetivo General   | Metodología   | Herramientas Sugeridas          | Tipo de Dato / Aplicación |
|----|---|--|---|---------------------------------|---------------------------|
| 1  | Monitorización estocástica de sistemas informáticos                   | Analizar patrones de fallos, errores y reinicios en sistemas de cómputo mediante modelos de Poisson y Markov.          | Recolección de logs, modelado probabilístico y simulación | Python (pandas, scipy.stats), R | Logs de sistemas          |
| 2  | Seguridad informática: detección estocástica de ataques e intrusiones | Modelar la ocurrencia de ataques o intentos de acceso mediante tema realacionados y procesos de Poisson no homogéneos. | Análisis de logs y simulación de tasas de ocurrencia      | Python (scipy, matplotlib)      | Datos de ciberseguridad   |
| 3  | Redes de computadoras : modelado de tráfico y solicitudes             | Estudiar el flujo de paquetes y conexiones en redes usando modelos de colas M/M/1 y procesos de Poisson.               | Captura de datos, modelado de colas y simulación          | SimPy, Wireshark, Python        | Datos de red              |
| 4  | Análisis estocástico de transacciones en bases de datos               | Modelar la llegada de transacciones, consultas y bloqueos para   | Simulación discreta de procesos de llegada y espera       | Python (SimPy), R (queueing)    | Datos de bases de datos   |

|    |  |   |  |                                    |                                    |
|----|--|---|--|------------------------------------|------------------------------------|
|    |  | optimizar el rendimiento.   |  |                                    |                                    |
| 5  | Sistemas distribuidos: sincronización y replicación aleatoria                        | Simular el comportamiento de mensajes y replicas en sistemas distribuidos con retardos estocásticos.            | Simulación de procesos y análisis probabilístico | Python (SimPy, multiprocessing)    | Datos de sistemas distribuidos     |
| 6  | Procesamiento por lotes programado: optimización estocástica de tiempos de ejecución | Modelar y simular la ejecución de tareas por lotes considerando incertidumbre en la duración.                   | Análisis de rendimiento y simulación Monte Carlo | Python (numpy, matplotlib)         | Datos de tareas programadas        |
| 7  | Carga y fallos de servidores en horarios laborales                                   | Analizar variaciones diarias en la carga del servidor y su relación con fallos mediante procesos no homogéneos. | Modelado temporal y simulación estocástica       | Python (pandas, statsmodels)       | Datos de rendimiento de servidores |
| 8  | Tráfico web con patrones diarios y semanales   | Modelar el tráfico de usuarios web considerando periodicidad y aleatoriedad en la demanda.                      | Serie temporales estocásticas y simulación       | Python (statsmodels), R (forecast) | Logs de tráfico web                |
| 9  | Modelos de Markov ocultos aplicados al reconocimiento de voz o texto                 | Implementar un modelo HMM para identificar patrones lingüísticos o fonéticos.                                   | Simulación y entrenamiento de modelos ocultos    | Python (hmmlearn), R               | Datos textuales o de audio         |
| 10 | Modelado de series temporales estocásticas para predicción de demanda                | Aplicar modelos ARIMA o Poisson para predecir tráfico o consumo.  | Modelado estadístico y validación                | Python (statsmodels), R            | Serie temporales                   |
| 11 | Simulación estocástica en  | Analizar la incertidumbre   | Experimentación con                              | Python (PyMC3,                     | Datos de predicción                |

|    |   |  |   |                              |   |
|----|---|--|---|------------------------------|---|
|    | redes neuronales bayesianas   | en modelos de aprendizaje automático.                                    | variabilidad aleatoria                                      | TensorFlow Probability)      |   |
| 12 | Análisis estocástico de fluctuaciones en criptomonedas                    | Simular precios y volatilidad con procesos de Wiener.                    | Simulación Monte Carlo y análisis de series                 | Python (NumPy, matplotlib)   | Datos financieros                                       |
| 13 | Modelos de colas en servicios en la nube                                  | Evaluar rendimiento y asignación de recursos en sistemas de colas M/M/c. | Simulación discreta y modelado                              | SimPy, R (queueing)          | Datos de servidores                                     |
| 14 | Procesos de nacimiento y muerte en gestión de errores de software         | Modelar la evolución de bugs y correcciones en sistemas de software.     | Análisis temporal de incidencias y defectos en repositorios | Python (matplotlib, seaborn) | Datos de desarrollo. Datos de software (GitHub, GitLab) |
| 15 | Simulación estocástica del flujo de estudiantes en plataformas e-learning | Analizar el comportamiento de acceso y permanencia.                      | Simulación y análisis de datos educativos                   | Python (pandas, matplotlib)  | Datos educativos  |
| 16 | Difusión de información en redes sociales con modelos de Poisson          | Estudiar la propagación de mensajes o fake news.                         | Modelado estocástico y visualización de redes               | Python (networkx, seaborn)   | Datos de redes sociales                                 |
| 17 | Desarrollo de un simulador interactivo de procesos estocásticos           | Crear un software educativo para visualizar Poisson, Markov y colas.     | Diseño de software y simulación numérica                    | Python (tkinter, matplotlib) | Simulación educativa                                    |
| 18 | Predicción de secuencias con cadenas de Markov en usuarios web            | Modelar la navegación de usuarios para anticipar comportamiento s.       | Análisis de datos reales y simulación de transiciones       | Python (Markovify, pandas)   | Logs de navegación web                                  |
| 19 | Modelos de colas M/M/1 para optimización                                  | Evaluar el rendimiento de sistemas bajo diferentes                       | Diseño experimental y simulación discreta                   | SimPy (Python), MATLAB       | Datos de red  |

|    |   |  |  |                              |                        |
|----|---|--|--|------------------------------|------------------------|
|    | de tráfico de red   | cargas.  |  |                              |                        |
| 20 | Procesos de Poisson en detección de ataques cibernéticos              | Analizar patrones de ocurrencia de intentos de intrusión.    | Análisis probabilístico y simulación           | Python (scipy.stats, pandas) | Logs de seguridad      |
| 21 | Movimiento Browniano en predicción de precios financieros             | Simular trayectorias aleatorias de precios de activos.       | Modelado estocástico y simulación Monte Carlo  | Python (NumPy, pandas)       | Datos financieros      |
| 22 | Simulación de cargas de trabajo en entornos distribuidos              | Evaluar desempeño bajo condiciones aleatorias de llegada.    | Modelos de colas y simulación                  | SimPy, AnyLogic              | Datos de rendimiento   |
| 23 | Modelos de Markov en secuencias genéticas                             | Analizar transiciones entre bases de ADN.                    | Modelado probabilístico y simulación           | Python (BioPython, NumPy)    | Datos biológicos       |
| 24 | Predicción estocástica de fallos en hardware                          | Estimar probabilidad de fallos bajo carga continua.          | Modelado de supervivencia y simulación         | Python (lifelines), R        | Datos de mantenimiento |
| 25 | Procesos de Poisson para mutaciones genéticas                         | Modelar ocurrencia de mutaciones en poblaciones.             | Modelado matemático y validación computacional | Python/R                     | Datos genéticos        |
| 26 | Desarrollo de un simulador en Python de procesos de Poisson y Markov  | Crear una herramienta educativa interactiva.                 | Diseño de software y simulación numérica       | Python (tkinter, matplotlib) | Simulación educativa   |
| 27 | Procesos de Wiener aplicados al modelado de temperatura ambiental     | Analizar fluctuaciones de temperatura a través del tiempo.   | Modelado de series temporales y simulación     | R, Python                    | Datos meteorológicos   |
| 28 | Procesos estocásticos aplicados a sistemas inteligentes de transporte | Simular llegadas y esperas en sistemas de transporte urbano. | Modelos de colas y análisis probabilístico     | Python, MATLAB               | Datos de movilidad     |
| 29 | Comparación entre modelos   | Evaluar ventajas del enfoque                                 | Análisis comparativo y                         | R, Python                    | Datos simulados        |

|  |   |                 |            |  |  |
|--|---|-----------------|------------|--|--|
|  | deterministas y estocásticos en predicción de datos | probabilístico. | simulación |  |  |
|--|---|-----------------|------------|--|--|

### **Estructura del documento a presentar**

- Introducción
- Planteamiento del problema
- Objetivo general y específicos
- Justificación
- Marco Teórico (conceptos teóricos sobre el tema de procesos estocásticos respecto al caso)
- Diseño metodológico (Definir el tipo de estudio (simulación, modelado, análisis de datos, herramientas empleadas y descripción del conjunto de datos en los escenarios simulados)
- Resultados esperados
- Bibliografía

### **Presentación Oral o Exposición**

Debe incluir los siguientes elementos:

- Introducción clara del problema y su contexto real.
- Fundamento teórico: tipo de proceso estocástico aplicado (Markov, Poisson, etc.).
- Modelo propuesto: diagrama, ecuación o simulación representativa.
- Demostración computacional: ejecución de un script o análisis de resultados.
- Resultados esperados o preliminares con interpretación.
- Conclusiones y posibles aplicaciones prácticas.
- Respuestas a preguntas del docente o compañeros.

### **Criterios de Evaluación**

- Claridad del problema y objetivos (Relevancia y coherencia entre el problema planteado y los objetivos formulados)
- Fundamento teórico y rigor científico (Dominio conceptual, uso de bibliografía actual y precisión en la exposición)
- Diseño metodológico (Correcta aplicación del modelo estocástico y justificación de herramientas)
- Creatividad y aplicación informática (Innovación en el uso de simulación, visualización o desarrollo de software)

- Presentación escrita y oral (Claridad, estructura y calidad del lenguaje técnico)
- Análisis crítico y argumentación (Capacidad de reflexión y solidez en la defensa del proyecto).