Trabajo Práctico Integral — Simulación

Maestría en Ciencia de Datos

Simulación y Optimización en Ciencia de Datos

Table of contents

1 Introducción

Este Trabajo Práctico (TP) integra los contenidos vistos en las clases:

- **Unidad I** – Generación de números pseudoaleatorios y Monte Carlo
- **Unidad II** – Bayes, cadenas de Markov y Metropolis–Hastings
- **Unidad III** – Simulación de eventos discretos (SED)
- **Unidad IV** – Procesos continuos (NHPP, CTMC, SDE)
- **Unidad V** – Reacciones químicas estocásticas: Gillespie SSA y Next Reaction Method

El objetivo es que el alumno implemente técnicas de simulación, compare métodos, valide sus resultados y presente visualizaciones claras.

Cada apartado incluye consignas, código sugerido y entregables obligatorios.

---

2 Parte 1 — Generación de Números Pseudoaleatorios y Monte Carlo

2.1 1.1 Implementación de un LCG

Implementar un **Generador Congruencial Lineal (LCG)**:

$$X_{n+1} = (aX_n + c) \mod m$$

2.1.1 Parámetros obligatorios

- Un set "bueno"
- Uno "malo" (ej., RANDU)

2.1.2 Código base (R)

```r
lcg <- function(n, seed = 123, a, c, m) {
  x <- numeric(n)
  x[1] <- seed
  for (i in 2:n) x[i] <- (a * x[i-1] + c) %% m
  return(x / m)
}
```

2.1.3 Código base (Python)

```python
def lcg(n, seed=123, a=1103515245, c=12345, m=2**31):
    x = seed
    seq = []
    for _ in range(n):
        x = (a * x + c) % m
        seq.append(x/m)
    return seq
```

2.1.4 Entregables

- Histogramas
- Runs Test
- Gráfico de triples (u_i, u_{i+1}, u_{i+2})

2.2 1.2 Estimación de por Monte Carlo

Simular puntos en el cuadrado $[0, 1]^2$ y estimar:

$$\pi \approx 4 \cdot \mathrm{mean}(x^2 + y^2 \leq 1)$$

2.2.1 Código base

```r
N <- 10000
u1 <- runif(N)
u2 <- runif(N)
pi_est <- 4 * mean(u1^2 + u2^2 <= 1)
```

---

3 Parte 2 — Metropolis–Hastings y Bayesian Inference

3.1 2.1 Posterior beta analítica

Sea una moneda con 10 lanzamientos y 7 caras.

$$\text{Posterior} = \text{Beta}(8, 4)$$

3.2 2.2 Implementación MH

3.2.1 Código base en R

```r
mh <- function(n_iter=5000, start=0.5, prop_sd=0.1){
  theta <- numeric(n_iter)
  theta[1] <- start
  for(t in 2:n_iter){
    prop <- rnorm(1, theta[t-1], prop_sd)
    if(prop < 0 || prop > 1){
      theta[t] <- theta[t-1]
    } else {
      num <- dbinom(7,10,prop) * 1
      den <- dbinom(7,10,theta[t-1]) * 1
      alpha <- min(1, num/den)
      theta[t] <- ifelse(runif(1) < alpha, prop, theta[t-1])
    }
  }
  theta
}
```

3.2.2 Diagnósticos obligatorios

- Traceplot
- Autocorrelación
- Histograma vs Beta(8,4)
- R-hat con 4 cadenas
- ESS

---

4 Parte 3 — Simulación de Eventos Discretos (M/M/1 o M/M/c)

Simular durante 8 horas un sistema de colas con:

- Llegadas Poisson: = 10 / hora
- Servicio exponencial: = 4 / hora
- Opcional: c servidores

Se debe implementar **LEF (Lista de Eventos Futuros)**.

4.1 Código base (R)

```r
simulate_mm1 <- function(lambda, mu, T_max){
  t <- 0
  n <- 0
  t_arr <- rexp(1, lambda)
  t_dep <- Inf

  arrivals <- 0
  departures <- 0
  wait_times <- c()

  while(t < T_max){
    if(t_arr < t_dep){
      t <- t_arr
      n <- n + 1
      arrivals <- arrivals + 1
```

```
      if(n == 1) t_dep <- t + rexp(1, mu)
      t_arr <- t + rexp(1, lambda)
    } else {
      t <- t_dep
      n <- n - 1
      departures <- departures + 1

      wait_times <- c(wait_times, t_dep)
      t_dep <- if(n > 0) t + rexp(1, mu) else Inf
    }
  }

  list(arrivals=arrivals, departures=departures, wait_times=wait_times)
}
```

---

5 Parte 4 — Modelos Continuos (Elegir uno)

5.1 Opción A: NHPP por Thinning

$$\lambda(t) = 5 + 3\sin\left(\frac{2\pi t}{24}\right)$$

Simular 7 días (168 horas).

5.2 Opción B: CTMC

Matriz de tasas:

From	To	Rate
A	B	0.4
A	C	0.2
B	A	0.1
C	A	0.3

Simular **2000 saltos**.

5.3 Opción C: SDE (GBM)

$$dS_t = 0.05S_t dt + 0.2S_t dW_t$$

Simular 1000 trayectorias.

---

6 Parte 5 — Gillespie SSA o Next Reaction Method

Sistema:

- $\rightarrow$ mRNA con tasa (k_1 = 10)
- mRNA $\rightarrow$ con tasa (k_2 = 1)

6.1 Código base SSA

```
gillespie <- function(Tmax, k1=10, k2=1){
  t <- 0
  X <- 0
  ts <- c(t)
  xs <- c(X)

  while(t < Tmax){
    a1 <- k1
    a2 <- k2 * X
    a0 <- a1 + a2

    r1 <- runif(1)
    r2 <- runif(1)

    tau <- -log(r1) / a0

    if(r2 * a0 < a1){
      X <- X + 1
    } else {
      X <- max(0, X - 1)
```

```
    }

    t <- t + tau
    ts <- c(ts, t)
    xs <- c(xs, X)
  }

  data.frame(time=ts, X=xs)
}
```

---

7 Parte Integradora (Opcional)

Combinar dos modelos (ejemplo):

- NHPP → pacientes
- M/M/1 → atención
- Gillespie → biomarcadores

Evaluar: **% de pacientes cuyo biomarcador excede un umbral antes de ser atendidos.**

---

8 Formato y entrega

- Informe PDF (generado desde este Quarto)
- Código R/Python
- Gráficos
- Discusión de supuestos
- Semillas fijas
- Comparaciones teóricas vs simuladas
- *Fecha de entrega: 20 de diciembre de 2025*

---

9 Criterios de Evaluación

Sección	Puntaje
Parte 1 – RNG + Monte Carlo	15
Parte 2 – MCMC MH	20
Parte 3 – Eventos Discretos	20
Parte 4 – Continuos	20
Parte 5 – SSA/NRM	20
Claridad y presentación	5