



**Nombre:**

Dianny Michelle De los santos De los santos

**Matrícula:**

2024-0213

**Docente:**

Wilmer Jose Fariña Piñero

**Asignatura:**

Programación Paralela

**Tema:**

Simulación Monte-Carlo de epidemias paralela

**Fecha:**

10/12/2025

# 1. Introducción

Este proyecto implementa un simulador epidemiológico basado en el modelo SIR (Susceptible-Infectado-Recuperado) sobre una grilla de  $1000 \times 1000$  celdas, representando 1 millón de individuos. Se desarrollaron versiones secuencial y paralela para analizar el desempeño mediante experimentos de escalabilidad fuerte con 1, 2, 4 y 8 núcleos de procesamiento.

## 2. Modelo SIR

### 2.1 Estados y Transiciones

El modelo divide la población en cuatro estados: Susceptible, Infectado, Recuperado y Fallecido. Las transiciones ocurren probabilísticamente cada día: **(1)** Infección con probabilidad  $1 - (1 - 0.30)^n$  donde  $n$  son vecinos infectados, **(2)** Recuperación con 10% diario, **(3)** Muerte con 1% diario.

### 2.2 Parámetros

Parámetro	Valor	Parámetro	Valor
Población	1,000,000	Tasa infección ( $\beta$ )	0.30
Infectados iniciales	10	Tasa recuperación ( $\gamma$ )	0.10
Duración	365 días	Tasa mortalidad ( $\mu$ )	0.01

El  $R_0$  (número básico de reproducción) se calcula como casos totales acumulados entre infectados iniciales.

## 3. Implementación

### 3.1 Arquitectura

Sistema modular con componentes: SIRModel (gestiona grilla y reglas), Statistics (recolecta datos), Visualizer (genera gráficas) y GridSimulator (orquesta simulación).

### 3.2 Versión Secuencial

Procesa la grilla celda por celda usando doble buffer para evitar inconsistencias. Se evalúa el estado actual, determina el próximo estado según reglas probabilísticas, y actualiza la grilla completa de forma atómica.

### 3.3 Versión Paralela

Emplea descomposición de dominio dividiendo la grilla en bloques. Para 8 núcleos se crean 9 bloques (3×3) de ~333×333 celdas. Cada núcleo procesa bloques independientemente, sincronizándose mediante barrera implícita. Las estadísticas se calculan con operaciones atómicas. Cada thread mantiene su generador aleatorio con semilla diferente.

## 4. Resultados

### 4.1 Comportamiento Epidemiológico

La simulación muestra cuatro fases: **(1)** Crecimiento exponencial (días 1-50): 10 a ~16,000 infectados, **(2)** Pico epidémico (días 50-150): máximo de ~40,000 casos, **(3)** Decaimiento (días 150-300): disminución por agotamiento de susceptibles, **(4)** Cola endémica (días 300-365): ~10,000-20,000 casos remanentes,  $R_0$  final ~90,000.

### 4.2 Comparación de Versiones

Métrica	Secuencial	Paralela (8 núcleos)	Diferencia
Susceptibles	74,683	73,625	1.4%
Recuperados	831,659	831,990	0.04%
Fallecidos	83,603	84,470	1.0%
Afectados	92.53%	92.64%	0.11%

Diferencias menores al 1.5% validan la correctitud, siendo atribuibles a la naturaleza estocástica del modelo.

### 4.3 Análisis de Desempeño

Núcleos	Tiempo (s)	Aceleración	Eficiencia	Núcleos	Tiempo (s)	Aceleración	Eficiencia
---------	------------	-------------	------------	---------	------------	-------------	------------

1	2.90	1.00×	100.0%	4	1.78	1.63×	40.7%
2	1.97	1.47×	73.7%	8	1.40	2.07×	25.8%

**Resultados:** Reducción del 52% en tiempo (2.90→1.40s) con 8 núcleos. Aceleración de 2.07× vs ideal de 8.00×. Eficiencia del 25.8%.

**Factores limitantes:** Overhead de sincronización (35%), contención de memoria (25%), desbalance de carga (20%), overhead de tasks (20%).

Según la Ley de Amdahl con fracción secuencial del 15%, el speedup máximo es 4.71×. Nuestro speedup (2.07×) alcanza el 44% de este límite.

## 5. Conclusiones

### Logros y Limitaciones

El proyecto implementó exitosamente dos versiones consistentes (diferencias <1.5%) logrando reducción significativa del tiempo. La descomposición de dominio es efectiva para problemas espacialmente estructurados. Sin embargo, la escalabilidad es sub-lineal (eficiencia 25.8% en 8 núcleos) por componentes secuenciales y costos de sincronización.