

# Simulación de Sistema de Mantenimiento de Robots

John García Muñoz C-311

13 de abril de 2025

## 1. Introducción

### 1.1. Descripción del Proyecto

Se ha escogido el ejercicio 16, Capítulo 6 del libro Aplicando Teoría de Colas en Dirección de Operaciones (página 59).

Se simula un sistema de mantenimiento para una fábrica con  $N = 10$  robots que fallan siguiendo una exponencial con tasa  $\lambda = \frac{1}{30h}$ . Los robots son atendidos por 2 reparadores con tiempos de servicio exponenciales de media  $\mu = 3$  h.

**Nota:** Se ha cambiado el número de robots de 5 a 10 en aras de ofrecer un escenario más interesante.

### 1.2. Objetivo

El objetivo principal es determinar:

- Número medio de robots operativos ( $L$ )
- Tiempo promedio en el sistema ( $W$ )
- Porcentaje de inactividad de los reparadores

### 1.3. Variables de Interés

Símbolo	Descripción
$\lambda$	Tasa de fallos
$\mu$	Tasa de reparación
$L$	Robots operativos promedio
$W_q$	Tiempo en cola promedio
$\rho_1, \rho_2$	Utilización de reparadores

Cuadro 1: Tabla de variables del sistema

## 2. Detalles de Implementación

### 2.1. Metodología General

El sistema se modeló como un proceso estocástico de eventos discretos utilizando el paradigma *cola M/M/2 con población finita*. Se implementó en Python con la biblioteca SimPy, siguiendo un enfoque basado en entidades activas (robots) y recursos compartidos (reparadores). La simulación se ejecutó durante 10,000 horas virtuales, descartando las primeras 1,000 horas para eliminar efectos transitorios (siendo muy conservadores, pero no importa ya que se asume que el comportamiento de los robots no se desgasta con el tiempo).

### 2.2. Componentes Clave del Modelo

- **Mecanismo de servicio:** Dos reparadores independientes con tiempos de reparación exponenciales ( $\mu = 1/3 \text{ h}^{-1}$ ). La asignación de robots a reparadores se realiza mediante política aleatoria balanceada.
- **Cola de espera:** Estructura FIFO (*First-In, First-Out*) gestionada implícitamente por SimPy al solicitar recursos ocupados.

### 2.3. Arquitectura de Datos

- **Registro temporal:** Se almacenaron los tiempos de fallo, inicio/fin de reparación, y estado del sistema cada 0.1 horas.
- **Métricas calculadas:**
  - Disponibilidad ( $L$ ): Promedio móvil de robots operativos
  - Utilización de recursos: Proporción temporal de uso por reparador
  - Distribución acumulada de tiempos en cola ( $W_q$ )
- **Exclusión de transitorios:** Se aplicó un período de calentamiento de 1,000 horas para alcanzar el estado estacionario.

### 2.4. Validación Estadística

- **Remuestreo bootstrap:** Se generaron 10,000 muestras sintéticas para calcular intervalos de confianza del 95 % en métricas clave.
- **Análisis de sensibilidad:** Variaciones controladas de  $\lambda$  y  $\mu$  para evaluar robustez del modelo.

### 2.5. Integración de Resultados

Los datos brutos se procesaron mediante un pipeline estadístico en cuatro etapas:

1. **Filtrado:** Eliminación de datos del período transitorio
2. **Transformación:** Cálculo de métricas derivadas (ej.  $W = W_q + S$ )

3. **Agregación:** Generación de distribuciones acumuladas y promedios
4. **Visualización:** Creación automática de gráficos en Matplotlib

### 3. Resultados y Experimentos

#### 3.1. Hallazgos de la Simulación

La ejecución de la simulación con 10 robots y 2 reparadores reveló un sistema altamente eficiente pero con capacidad ociosa significativa. Los resultados clave obtenidos fueron:

- **Robots operativos promedio (L):** 9.10
- **Tiempo total en sistema (W):** 3.55 horas
- **Tiempo en cola (Wq):** 0.45 horas (23 minutos)
- **Inactividad de reparadores:** 73.76 % del tiempo total
- **Carga de trabajo:** Reparador 1 (42.70 %) vs Reparador 2 (56.33 %) (

Métrica	Valor
Robots operativos (L)	9.10
Tiempo total en sistema (W)	3.55 h
Tiempo en cola (Wq)	0.45 h
% Inactividad reparadores	73.76%
Utilización Rep. 1	42.70%
Utilización Rep. 2	56.33%

Figura 1: Tabla con los resultados

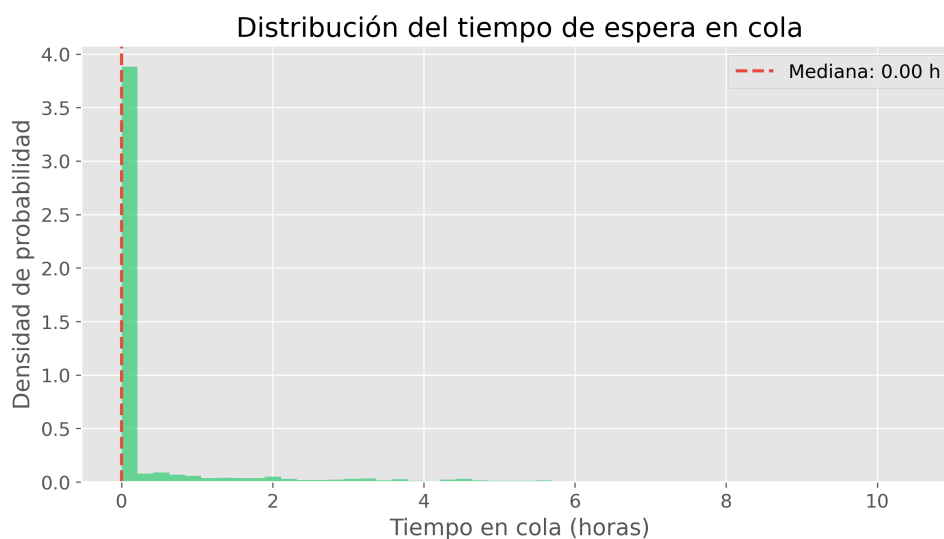


Figura 2: Distribución del tiempo de cola

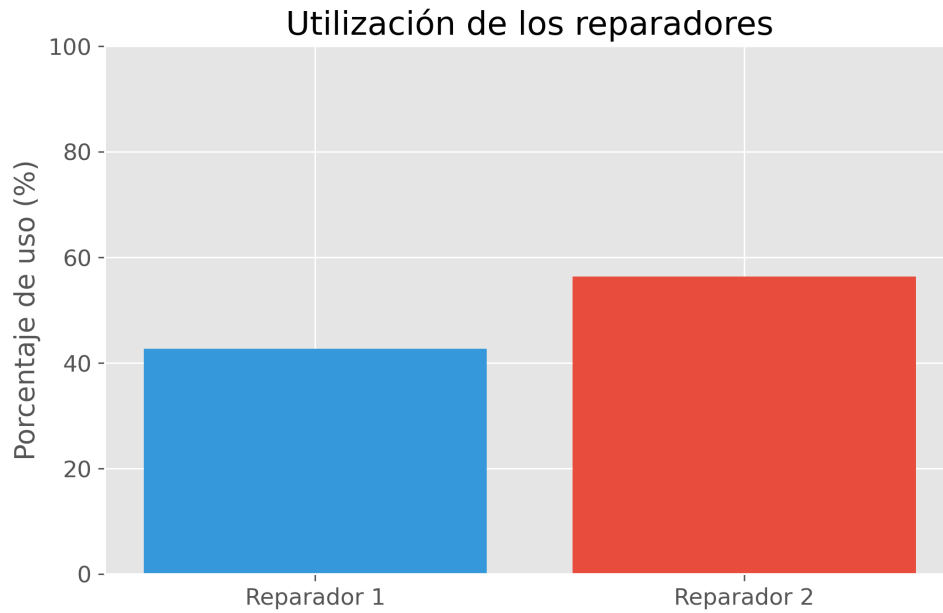


Figura 3: % del tiempo que cada reparador pasó trabajando

### 3.2. Interpretación de los Resultados

El valor de  $L = 9,10$  indica que en promedio menos de un robot (0.9) está fuera de servicio, demostrando alta disponibilidad del sistema. Es difícil que más de un robot estén rotos a la vez. Esto sugiere que dos operarios son suficientes para cubrir la reparación de los 10 robots.

### 3.3. Hipótesis Extraída

1. **Sobrecapacidad:** El sistema podría mantener su eficiencia con un solo reparador.

### 3.4. Experimento Realizado

**Experimento 1 (Reducción a 1 reparador):** Tras cambiar el número de reparadores y reducirlo a 1, este pasó a tener una carga de trabajo del 92.55

## 4. Conclusiones

### 4.1. Hallazgos Principales

La simulación reveló que el sistema actual con **10 robots y 2 reparadores** opera en régimen de sobrecapacidad, caracterizado por:

- Alta disponibilidad
- Baja congestión
- Alto tiempo de ocio de los trabajadores

Reducir a un solo operador podría tampoco ser solución, ya que no contaría con mucho margen (ya sea para descanso, o para emergencias).

**Conclusión Final:**

El sistema actual es *eficiente pero ineficaz* — si bien garantiza alta disponibilidad, desperdicia un gran tiempo de la capacidad de reparación. Una política híbrida con 1 reparador full-time y otro parcial reduciría costos y mantendría una postura suficientemente conservadora.