

1. Diseño de la idea: Parque de diversiones

Descripción de la simulación

Se simula un parque de diversiones durante un cierto número de minutos.

Tres actividades ocurren al mismo tiempo:

- a. **Taquilla:** vende entradas a todos los visitantes.
- b. **Atracciones** (Montaña Rusa, Carrusel, Casa del Terror): cada minuto atienden personas, pueden fallar y afectan la satisfacción.
- c. **Mantenimiento/Limpieza:** reduce la probabilidad de fallas de las atracciones.

Al final, se muestran métricas como ingresos, visitantes atendidos, satisfacción acumulada y fallas.

Qué va en paralelo y cómo se usan las cláusulas

- **sections:**
 - Sección 1: Taquilla
 - Sección 2: Operación de atracciones
 - Sección 3: Mantenimiento

Cada una es independiente y ocurre de forma simultánea.
- **parallel for:**
 - En Taquilla: recorrer visitantes.
 - En Operación: recorrer atracciones cada minuto.
 - En Mantenimiento: recorrer atracciones para aplicar mejoras.
- **firstprivate:** semilla de números aleatorios distinta por hilo para simular capacidades y fallas.
- **shared:** parámetros globales como `N_visitantes`, `precio_ticket`, `capacidad_atr`, `prob_falla_base`.
- **reduction:** para acumular ingresos, visitantes atendidos, satisfacción total y fallas totales sin usar `critical`.
- **critical:** solo para modificar variables compartidas que no se pueden reducir, como la probabilidad de falla.

Variables compartidas y privadas

Tipo	Variables
shared	N_visitantes, precio_ticket, capacidad_atr, prob_falla_base
shared con reduction	ingresos_totales, atendidos_totales, satisfaccion_total, fallas_totales
firstprivate	seed para el generador de números aleatorios
private	índices de bucle (i, j, k), variables temporales (served, fallo, delta_satisf, mejora)

2. Pseudocódigo**INICIO**

Definir N_VISITANTES, MINUTOS_SIMULACION, NUM_ATRACCIONES

Definir PRECIO_TICKET

Definir CAPACIDAD_ATRACCION[NUM_ATRACCIONES]

Definir PROB_FALLA_BASE[NUM_ATRACCIONES]

Definir ingresos_totales = 0 // reduction

Definir atendidos_totales = 0 // reduction

Definir satisfaccion_total = 0 // reduction

Definir fallas_totales = 0 // reduction

EJECUTAR EN SECCIONES PARALELAS:

SECCIÓN 1: Taquilla

PARA CADA visitante EN paralelo (parallel for, firstprivate semilla)

sumar PRECIO_TICKET a ingresos_totales

aumentar atendidos_totales en 1

SECCIÓN 2: Operación de atracciones

PARA minuto = 1 HASTA MINUTOS_SIMULACION

PARA CADA atracción EN paralelo (parallel for, firstprivate semilla)

calcular personas_atendidas con variación aleatoria

determinar si hay falla usando PROB_FALLA_BASE

Sofia Garcia - 22210

Julio Garcia Salas - 22076

SI hay falla

 aumentar fallas_totales

 restar satisfacción

SINO

 sumar satisfacción

SECCIÓN 3: Mantenimiento

 PARA ronda = 1 HASTA RONDAS_MANTENIMIENTO

 PARA CADA atracción EN paralelo (parallel for, firstprivate semilla)

 calcular mejora aleatoria

 EN ZONA CRÍTICA (critical)

 reducir PROB_FALLA_BASE de esa atracción

MOSTRAR resultados finales:

 ingresos_totales, atendidos_totales, satisfaccion_total, fallas_totales

 mostrar PROB_FALLA_BASE final por atracción

FIN