

## 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:

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

Al final, se muestran métricas como ingresos, visitantes atendidos, satisfacción acumulada, fallas totales y un desglose por atracción (fallas y satisfacción por atracción). También se imprime el número máximo de hilos usados para evidenciar el paralelismo.

### 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 y arreglos de desglose fallas\_por\_atr, satisf\_por\_atr.

Sofia Garcia - 22210

Julio Garcia Salas - 22076

- **reduction:** para acumular ingresos, visitantes atendidos, satisfacción total y fallas totales sin usar critical.
- **critical:**
  - Para modificar variables compartidas que no se pueden reducir, como la probabilidad de falla.
  - Para fusionar los acumuladores de desglose por atracción (fallas\_por\_atr, satisf\_por\_atr).

### Variables compartidas y privadas

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

### 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

Sofia Garcia - 22210

Julio Garcia Salas - 22076

Definir fallas\_por\_atr[] = 0      // shared, actualizado en critical

Definir satisf\_por\_atr[] = 0      // shared, actualizado en critical

Mostrar número máximo de hilos

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

SI hay falla

aumentar fallas\_totales y fallas\_por\_atr[atracción]

restar satisfacción y acumular en satisf\_por\_atr[atracción]

SINO

sumar satisfacción y acumular en satisf\_por\_atr[atracción]

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

mostrar fallas\_por\_atr y satisf\_por\_atr

FIN