

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



## Mini-Mini Proyecto

Paralelizacion de simulador de trafico

José Rodrigo Marchena, 22398

Juan Luis García Zarceño  
Programacion Paralela  
Guatemala, 2025

## Repositorio

[https://github.com/MarchMol/omp\\_traffic\\_sim](https://github.com/MarchMol/omp_traffic_sim)

### Descripción del problema

El objetivo de este mini proyecto es desarrollar una solución para la simulación de tráfico usando agentes que representan vehículos, semáforos e intersecciones. Esta debe ser paralelizada y justificada con OpenMP de manera que la actualización de los datos se ejecute de manera simultánea y que provea una ventaja de tiempo con respecto a una versión secuencial del mismo algoritmo. Cada agente también, deberá poseer acciones como cambiar de estado para un semáforo, o avanzar en una intersección para automóviles. Se incentiva la evaluación de diferentes abordajes de paralelización para poder encontrar una topología que optimice el uso de recursos.

### Estructura definida

La estructura definida para almacenar los valores de automóviles, semáforos e intersecciones permite la paralelización de tareas al mantener información independiente. Se plantean 2 estructuras que agrupan la información: Roads e Intersections

#### - **Defined:**

- Constantes de simulación:
  - **NUM\_CARS:** cantidad de carros totales (en teoría)
  - **NUM\_INTERSECTION:** cantidad de roads o calles que puede tener una intersección
  - **CAR\_PER\_INT:** cantidad de carros que puede tener una intersección
  - **NUM\_SIMULATIONS:** cantidad de simulaciones a evaluar.
- Constantes de semáforo:
  - **RED\_LIGHT**
  - **YELLOW\_LIGHT**
  - **GREEN\_LIGHT**

#### - **Road:**

- Parámetros:
  - **Num\_Cars:** cantidad de carros en la calle (para facilitar iteración)
  - **Car\_Pos:** lista que contiene las posiciones de los carros en dicha calle.
  - **Car\_idx\_start:** se asume que cada calle o Road tiene una sección continua de los carros totales, tomando como inspiración la paginación en sistemas operativos, se almacena entonces de donde comienza y cuantas entradas le continúan; así se pueden diferenciar los carros de diferentes calles.
  - **State:** estado del semáforo, ya sea luz roja, verde o amarilla.

- **Intersection:**
  - Parámetros:
    - **Num\_Roads:** cantidad de calles que tiene una intersección
    - **Roads:** lista de las diferentes calles que pueden pertenecer a una intersección.

Para fines de este proyecto, se tomara sola una intersección que contiene variables cantidades de calles, aunque el código es escalable para tomar en cuenta múltiples intersecciones.

### *Lógica de simulación*

La lógica consiste en las funciones para actualizar los semáforos y las posiciones de los carros

- **Actualización de semáforos:**
  - Se iteran sobre todas las Roads de una Intersección y conociendo que cada una solo posee un semáforo, se aplica la conversión apropiada. No hay tiempos de espera para las luces, solo siguen el orden de rojo lleva a verde, verde lleva a amarillo y amarillo lleva a rojo.
- **Actualización de posiciones**
  - Se iteran sobre todas las roads posibles de una intersección y primero se revisa si el semáforo no es rojo, y si esta condición se cumple, se continua a sumar 1 a la posición de todos los carros de esa Road.

### *Estrategia de paralelización*

Dentro del bucle de simulación, dado que este se debe ejecutar secuencialmente, se inicializa una sección paralela dentro de él. Asimismo, se introduce el contenido de las funciones que actualizan las posiciones de los carros y los semáforos y estos se establecen dentro de un `parallel for`. En este, debido que por ejemplo, hay carros que no avanzaran porque tienen una luz roja, o semáforos que tienen que pasar por mas comprobaciones, se observo que el uso de una tag `nowait` llevo a mejores tiempos de ejecución.

Asimismo, se considero innecesaria la implementación de dinamismo en estos bucles paralelizados, pues simplemente conllevaban mas uso de recursos y no optimizaban el tiempo. Se mantuvo una calendarización estática para ambos bucles.