

Instituto Tecnológico y de Estudios Superiores de Monterrey  
Campus Monterrey



TC2005B - Construcción de software y toma de decisiones  
Grupo: 503

### **M5. Revisión de avance 1**

#### **Profesores:**

Alfredo Alan Flores Saldivar  
Raúl Valente Ramírez Velarde

#### **Equipo #1**

Humberto Genaro Cisneros Salinas - A01723264

Jorge Adrián De la Garza Flores - A00838816

Alejandra Saldaña Rodríguez - A00838889

German Cueto Toledo - A00838445

13 de agosto del 2025

## **Reto elegido a trabajar: Tráfico vehicular en la ciudad**

Es importante para las personas que residen o trabajan en una ciudad llegar a su destino de manera eficiente y cómoda. Por eso, en una ciudad grande, la cantidad de vehículos transportándose y encontrándose en intersecciones (algunas de éstas con semáforos) lleva a un problema de alto tráfico, llevando a las personas a frustrarse por los largos tiempos de trayecto, impidiendo incluso llegar puntualmente a sus respectivas citas (Cuerpo de profesores de TC2008B, 2025)

(Es importante mencionar que la selección de este reto es temporal en lo que obtiene más información sobre el reto relacionado con SEMEX)

### **Objetivo de la Investigación**

En esta investigación se introduce el concepto de una simulación multiagente y su abreviación en inglés, ABM, debido al término “Agent Based Modeling”. Se enlistan algunos de los beneficios que pueden tener este tipo de simulaciones en el ámbito urbanístico de manera general y en el tráfico de una ciudad de una forma más específica. Con esta investigación se pretende construir las bases al concepto, mostrar ejemplos de proyectos que han tenido un impacto como aquel que se busca tener con este reto y finalmente dar un entendimiento general de lo que conlleva realizar una simulación en donde se une la modelación en un lenguaje de alto nivel y un visualizador. Esto último sembrará las bases para el desarrollo de una solución al reto en Python y Unity.

## **Investigación de la Movilidad Urbana**

### **¿Qué es una simulación multi-agente (ABM) aplicada al transporte?**

En una ABM de movilidad, “agentes” (personas, vehículos, semáforos, empresas de ride-hailing, etc.) toman decisiones individuales (rutas, modo, hora de salida, velocidad, aceptación de viajes) y sus interacciones producen los patrones de tráfico observados. A diferencia de modelos agregados, las ABM permiten representar heterogeneidad, aprendizaje, congestión dependiente del tiempo y retroalimentaciones entre oferta y demanda. Revisiones recientes destacan su utilidad para planificación, evaluación de políticas y estudio de tecnologías emergentes (vehículos autónomos, micromovilidad, MaaS).

### **¿En qué problemas urbanos ayuda?**

- Reducción de congestión y tiempos de viaje.
- Evaluar escenarios “qué pasaría si...”: peajes de zona, carriles bus-bici, cambios de fases semafóricas, gestión de carga/descarga. ABM permite medir impactos desagregados (por zona, corredor, tipo de usuario) y evitar efectos rebote.
- Probar alternativas de red (por ejemplo, agregar un carril bus o una ciclovía protegida) antes de construir, observando cambios en elecciones de ruta y modos. Herramientas como A/B Street están pensadas justo para explorar estas transformaciones urbanas.
- Gestión de transporte público y coordinación con la red vial.
- Con datos GTFS y la geometría vial (OpenStreetMap), es posible simular puntualidad, acumulación en paradas y efectos de prioridad semafórica sobre buses. GTFS y OSM son los estándares abiertos más comunes para alimentar estos modelos.
- Evaluación de servicios bajo demanda (TNC/ride pooling) y flotas autónomas.
- Las ABM permiten despachar vehículos, emparejar viajes, y medir ocupación y espera bajo competencia o cooperación entre operadores. Estudios muestran que la fragmentación entre múltiples operadores puede degradar el pooling si diluye la masa crítica.
- Simuladores microscópicos (p. ej., SUMO) se conectan con módulos de emisiones/energía para estimar CO<sub>2</sub>, consumo y beneficios de límites de velocidad o gestión de tráfico.

## **Calibración y validación**

- Calibración de demanda: ajustar parámetros de utilidad/elección (modo, hora, destino) para reproducir distribuciones objetivo (longitud de viaje, matriz OD, participaciones modales).
- Validación operacional: comparar volúmenes por enlace, velocidades por corredor, puntualidad de buses, y colas en intersecciones, contra sensores, aforos y AVL/RT de la agencia.
- Datos masivos para ajuste: uso de datos móviles y técnicas de asimilación (p. ej., EnKF) para calibración dinámica; estudios muestran reducciones de sesgo en matrices OD y distribución de longitudes de viaje.
- Ejemplo: Barcelona con MATSim alimentado por registros móviles para analizar un peaje cordón; se validó la calidad de los diarios reconstruidos y se evaluaron impactos por zona y viajero.

## Metodología práctica

Se define una pregunta de la política, se construye el escenario base, se genera una población sintética y se calibra después de haber obtenido datos según una cierta escala. Después se diseñan los escenarios, se corren las simulaciones y se obtienen KPIs (datos claves) con los que se pueden tomar decisiones sobre las modificaciones que se deben realizar para mejorar la situación que se está analizando.

## Casos ilustrativos

- Peaje urbano (política de gestión de demanda): Con MATSim + datos móviles, puedes estimar cambios de modo/hora y redistribución espacial del tráfico, además de ingresos por peaje y efectos fuera del cordón. (Caso Barcelona).
- Prioridad semafórica a buses (operación): Con SUMO/CityFlow puedes probar planes de señal y medir su efecto en buses y autos (tiempos y colas), e incluso entrenar controladores con RL para horarios pico variables.
- Ride-pooling multi-operador (mercados de movilidad): ABM permite simular despacho y emparejamiento bajo competencia vs. cooperación y medir el umbral de masa crítica para que el pooling funcione.
- Logística urbana / reparto nocturno: ABM integrando demanda y operaciones muestra impactos en VMT/energía y en ventanas de entrega a nivel metropolitano (Chicago con POLARIS).

## Buenas prácticas y cautelas

- Calibración/validación primero: un ABM sin buen ajuste puede dar respuestas precisas a la pregunta equivocada. Usa múltiples fuentes (aforos, AVL, móviles) y documenta supuestos.
- Ética y privacidad: si empleas telefonía móvil, respeta la regulación (anonimización, síntesis de movilidad) y minimiza riesgos (p. ej., generar datasets sintéticos).
- Computación: para escenarios grandes, necesitarás paralelismo/HPC (POLARIS) o ejecuciones nocturnas; para iteración rápida en cruces específicos, usa SUMO/CityFlow.
- Interoperabilidad: mantén pipelines reproducibles (config + seeds + versiones de datos) y publica escenarios “base” y “política” para auditoría.

## **Integración entre el diseño de agentes y la visualización en entornos como Unity**

Un aspecto crítico en la simulación de sistemas multiagentes para movilidad urbana es el diseño y la programación de los agentes, así como la conexión fluida de esta lógica con entornos de visualización avanzada, como Unity.

En un flujo de trabajo completo, los agentes no son simples puntos o símbolos sobre un plano; representan entidades autónomas con percepciones, objetivos y capacidades de decisión. Para modelar este comportamiento, es común utilizar lenguajes de programación de alto nivel como Python, C# o Java, que permiten:

- Definir reglas de decisión basadas en datos reales (por ejemplo, patrones de tráfico históricos o encuestas de movilidad).
- Implementar sistemas de percepción y respuesta (por ejemplo, detección de congestión y elección de rutas alternativas).
- Simular comportamientos heterogéneos, donde distintos agentes tengan prioridades y limitaciones distintas (automóviles privados, transporte público, bicicletas, peatones).

### **Conexión con entornos de visualización**

Al separar la lógica del sistema multiagentes de la representación gráfica, es posible probar, optimizar y validar el modelo sin la carga computacional que implica renderizar la simulación en tiempo real. Esto facilita:

- Escalabilidad en la simulación: Se pueden correr millones de iteraciones para calibrar el modelo antes de mostrarlo visualmente.
- Flexibilidad en la validación: El motor de agentes puede exportar datos en formatos como JSON o CSV, que luego pueden ser usados por distintas plataformas de visualización.
- Optimización del rendimiento: Los cálculos de comportamiento pueden ejecutarse en servidores o clústeres de cómputo, mientras que Unity se encarga únicamente de renderizar el resultado.

## **Por qué es importante separar la lógica de la visualización**

Separar el motor de simulación (lógica multiagente) del motor gráfico (visualización) permite integrarlos mediante interfaces bien definidas. Esto es crucial porque:

- Los algoritmos de decisión y aprendizaje de los agentes requieren precisión y control temporal, mientras que la visualización se enfoca en realismo visual y experiencia de usuario.
- La visualización puede requerir interpolación de posiciones y animaciones, lo cual no afecta la lógica interna de la simulación pero sí influye en cómo se presenta a un público no técnico (por ejemplo, autoridades municipales).
- Mantener esta separación mejora la modularidad del proyecto: el motor de agentes se puede actualizar o reemplazar sin rehacer toda la parte gráfica, y viceversa.

En la práctica, esta arquitectura modular se utiliza en proyectos como CityFlow y SUMO-Unity Bridge, donde el motor de simulación genera la dinámica de tráfico y Unity se encarga de mostrar el flujo vehicular en entornos 3D inmersivos para análisis y comunicación de resultados.

## **Conclusión**

- Las ABM permiten medir impactos detallados por agente y lugar, capturando cambios de comportamiento ante políticas de infraestructura, operación y precio.
- Plataformas maduras (MATSim, SUMO, POLARIS) y datos abiertos (OSM, GTFS) hacen viable su adopción por ciudades y universidades.
- Con calibración rigurosa (incluyendo datos móviles cuando sea posible), las ABM se convierten en un laboratorio virtual para reducir congestión, mejorar el transporte público, y evaluar nuevas movilidades con transparencia y trazabilidad científica.

## **Prompts Utilizados (ChatGPT 5):**

- Eres un investigador profesional del área de urbanismo y de computación con un doctorado en urbanismo y modelación computacional. Presenta una investigación respaldada por fuentes confiables de información y proyectos anteriores de cómo puede la simulación de multiagentes ayudar los problemas de movilidad urbana. Haz la investigación en un nivel de entendimiento de un alumno de computación de universidad.
- Agrega una sección a la investigación que hable sobre cómo es importante poder diseñar a los agentes, modelar el comportamiento de estos utilizando lenguajes de programación de alto nivel y conectando esta lógica directamente con un software de visualización como Unity para tener una solución completa. Concéntrate especialmente en por qué es diferente separar el sistema multiagentes de las gráficas computacionales.
- Dame todas las fuentes que utilizaste para la investigación en formato apa 7

## **Lo que se modificó:**

Se quitaron las secciones de proyectos anteriores y de pila tecnológica ya que la investigación era más cualitativa y no tan técnica. También se removieron puntos que no estaban correctamente explicados o que se repetían con otras secciones de la investigación. La parte de la metodología práctica fue escrita en muchos puntos que eran muy específicos y no estaban bien desarrollados. Se decidió resumir los puntos en un párrafo más general y relevante a la investigación.

Se hizo el segundo prompt para poder ligar mejor los conceptos con la clase y poder tener una mejor idea del diseño de las soluciones separando el desarrollo de sistemas multiagentes con la visualización. Para la penúltima sección, la IA utilizó mucho la negación a asunciones que no se habían hecho previamente en el texto como decir “no sólo Unity” cuando en ningún momento se menciona que solo se puede usar Unity. Se borraron estas negaciones y momentos en los que se incluía unity como visualizador sin previa explicación.

Finalmente, se removieron los links de las fuentes que se tenían dentro del texto y se dejaron solamente las referencias en formato APA 7 después de revisar que fueran verídicas y relevantes.

## Conformación del equipo:

### Humberto:

- **Fortalezas:** Organización, experiencia con programación en Python, experiencia con desarrollo en Unity y programación en C#.
- **Áreas de oportunidad:** No me acuerdo mucho de los temas de matemáticas, pero para mí es fácil recordar los conceptos, solo hace falta repasarlos.
- **Expectativas:** Quiero tener una solución completa que ayude a mejorar mi experiencia en la planeación y desarrollo de algo que ayude a una problemática. Quiero mejorar mis habilidades con Python y seguir extendiendo mi conocimiento en visualizadores como Unity.
- **Logros esperados:** Espero crear una simulación de tráfico cuyo funcionamiento pueda retratar correctamente la realidad y se puedan conseguir datos relevantes. En general, que la experiencia me ayude a replicar este tipo de proyectos con agentes y en donde se combinan varias tecnologías como Python y Unity en un ambiente laboral.
- **Compromisos:** Estoy comprometido a cumplir con las tareas que se me sean asignadas, apoyar con la organización y planeación y apoyar a mis compañeros con todo lo que necesiten a lo largo del proyecto.

### Alejandra:

- **Fortalezas:** Tengo experiencia en programación en diferentes lenguajes, al igual que en Unity. Disfruto aprendiendo nuevos temas que llaman mi atención.
- **Áreas de oportunidad:** A veces me cuesta concentrarme por mucho tiempo y pospongo algunas tareas, sin embargo siempre termino a tiempo y de manera correcta.
- **Expectativas:** Poder entender cómo funcionan los agentes y poder realizar mis propias simulaciones. Mejorar mis habilidades de programación y reforzar mis conocimientos matemáticos.
- **Logros esperados:** Desarrollar una simulación de tráfico que genere resultados útiles. Igualmente espero obtener el conocimiento sobre sistemas multiagentes para poder aplicarlo en proyectos futuros.
- **Compromisos:** Me comprometo a apoyar a mi equipo en lo que sea necesario y a cumplir con mis partes del proyecto en tiempo y forma. Igualmente me comprometo a investigar un poco más allá de lo que se requiera para tener un mejor entendimiento de los temas.

### Jorge:

- **Fortalezas:** Me gusta la parte de la programación, el hacer el código, al igual que buscar la mejor manera de hacer algo, más óptimo o eficiente, o cómo mejorar el proyecto para que llegue a un alcance más allá.



- **Áreas de oportunidad:** Mi organización es un poco complicada, tengo un horario un poco lleno y se me dificulta trabajar extensamente entre semana.
- **Expectativas:** Espero aprender mucho acerca de cómo funcionan los agentes, como programarlos, y manejarlos para que se comporten de manera correcta, y con ello poder hacer simulaciones. Se me hace una área de mucha oportunidad para muchos proyectos que pueden traer valor a la sociedad.
- **Logros esperados:** Crear una simulación del tráfico que sea fiel a la realidad, y poder extraer de ella una conclusión aplicable en la vida real de cómo podemos mejorar el flujo vehicular en nuestra ciudad o en la zona.
- **Compromisos:** Aprender habilidades de manejo de agentes y su integración en Unity, hacer uso de material externo a la clase para poder lograr nuestro objetivo al complementarlo con lo que aprendemos en clase, y poder colaborar para crear un producto terminado.

#### German:

- **Fortalezas:** Enfoque en encontrar la solución al problema planteado y explicarlo al igual que la solución. Conocimiento básico de Unity en proyectos 3D.
- **Áreas de oportunidad:** Pasar el pensamiento obtenido a un código de Python, para que sea aplicable.
- **Expectativas:** Manipular la creación de agentes y entender su comportamiento en un entorno diseñado.
- **Logros esperados:** Que los agentes actúen conforme su programa lo indica y que en la simulación se observen estos cambios en tiempo real.
- **Compromisos:** cumplir en tiempo y forma con mis delegaciones en el trabajo, dependiendo el área que me toque desarrollar. Además de mostrar intereses en la parte expositiva, formulando la mejor manera de explicar nuestra solución.

#### Descripción del reto a desarrollar

Se busca crear una simulación de multiagentes en la cual, cada agente controla un vehículo en las vialidades de una ciudad, interactuando entre sí con comportamientos como cruces con intersecciones, semáforos, señalamiento de alto o baja velocidad, curvas, cambio de carriles con indicación de direccionales, entre otras variables que involucren el movimiento en la urbe.

## **Identificación de los agentes involucrados**

Vehículos tipo Sedan: Son los más comunes, cuentan con una velocidad promedio de entre 60-120 km/h. Tienen una extensión promedio de 5 metros.

Vehículos tipo SUV: Son igual de comunes que los Sedan, sin embargo al ser más pesados son ligeramente más lentos, velocidad promedio de 60-100 km/h, en extensión son medio metro más largos.

Autobuses: Paradas continuas en un horario específico o aproximado. Menor velocidad de 60-80km. Tienen una extensión de 3 vehículos tipo Sedan.

Semáforos: Fijos, tienen 3 estados (verde, amarillo y rojo) que cambian con el tiempo, cada estado tiene un tiempo diferente y debe estar sincronizado con los demás.

Taxis: Manejan por el carril del costado y tienen paradas ocasionales. Mismo modelo físico de un tipo Sedan, solo cambia su comportamiento en manejo, suelen manejar por los carriles de baja velocidad puesto que se detienen frecuentemente.

Camiones de carga: Son los vehículos más grandes, por ende los más lentos, tienen aproximadamente una longitud de 4 carros tipo Sedan. En promedio van entre 40-60 km/h.

Motocicletas: Son vehículos compactos y suelen cambiar de carril frecuentemente. Tienen una extensión de 2.5 metros y van a una velocidad promedio de entre 60-120 km/h.

Scooters: Son vehículos especiales ya que no siguen las reglas convencionales del tráfico y se tienen que desplazar en el carril de baja velocidad. Tienen una extensión de 1.5 metros y su velocidad ronda entre los 20-60 km/h.

## Diagrama de clase presentando los distintos agentes involucrados.



Los agentes que se utilizaran son 3:

- Vehículos que incluye carros, camiones, vehiculos de carga, camionetas, entre otros
- Semáforo que controlan el flujo del tráfico
- Scooter que se comportan diferente a un vehículo tradicional, ya que se tienen que mantener en el carril de baja.

Estas clases son las principales, y de ahí pueden surgir subclases si es necesario para manejar los diferentes tipos de vehículos, semáforos, y scooters/bicis

## Diagrama de protocolos de interacción.

Vehículo → Semáforo:

El Vehículo le solicita el paso al semáforo para ver si puede cruzar

Semáforo → ControlCentral:

El semáforo le pregunta al control central de la simulación si es que puede autorizar el paso.

ControlCentral → Semáforo:

El control central ve el flujo vehicular y en base a ello acepta o deniega al semáforo el cambio de estado para que crucen los vehículos.

Semáforo → Vehículo:

El semáforo toma la acción y pone un tiempo para el cual la acción es válida.

Vehículo → Semáforo:

El vehículo toma la acción que le da el semáforo y continúa con su camino o espera a que sea aceptado.

## Plan de Trabajo

### Actividades Realizadas:

Actividad	Responsable(s)	Fecha Límite	Esfuerzo (1-10)
Crear Repositorio de GitHub	Humberto	13 de agosto, 2025	1
Hacer y corregir Investigación Guiada por IA	Humberto	14 de agosto, 2025	6
Hacer Presentación de Investigación	Humberto	14 de agosto, 2025	5
Video de diapositivas 1-3	Humberto	14 de agosto, 2025	6
Video de diapositivas 4-5	Alejandra	14 de agosto, 2025	6
Video de diapositivas 6-7	Jorge	14 de agosto, 2025	6
Video de diapositivas 8-9	Germán	14 de agosto, 2025	6
Objetivo de la Investigación	Alejandra	14 de agosto, 2025	4
Conformación del Equipo	Todos los miembros	14 de agosto, 2025	4
Crear Grupo de Whatsapp para Comunicación	Humberto	13 de agosto, 2025	1
Descripción del reto	Jorge	14 de agosto, 2025	5
Identificación de Agentes	German, Jorge y Alejandra	14 de agosto, 2025	7
Diagrama de clase	Jorge	14 de agosto, 2025	6
Diagrama de Protocolo	Jorge	14 de agosto, 2025	6

### Actividades Pendientes:

Actividad	Tiempo Estimado de Realización
Listado de requerimientos para la simulación en base a diagramas	2 días
Modelación de los agentes con AgentPy	2 días
Diseño del entorno de la simulación	3 días
Diseño del modelo en el que interactúan los agentes con el entorno.	4 días
Implementación de la simulación completa en Python	5 días
Selección de diseño y/o modelos existentes de las entidades en Unity	3 días
Implementación del sistema de visualización en Unity	5 días
Diseño de la comunicación entre Python y Unity	2 días
Implementación de comunicación entre la simulación y el visualizador (Python con Unity)	1 día (Exámen Integrador)

## **Enlaces a Contenido Adicional**

Link al Repositorio de Github: <https://github.com/EliteGentro/TC2008B.103>

Link a la Presentación:

[https://www.canva.com/design/DAGv\\_hSqJuw/-eraQM23VbXmYjY6R\\_huOw/edit?utm\\_content=DAGv\\_hSqJuw&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGv_hSqJuw/-eraQM23VbXmYjY6R_huOw/edit?utm_content=DAGv_hSqJuw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Link al Video:

<https://drive.google.com/file/d/1n-mFFWV1CFF58XOi32Rme1PMzeXchWG2/view?usp=sharing>

## Referencias:

1. **A/B Street (plataforma colaborativa)**  
A/B Street. (s. f.). *Project history – A/B Street*. Recuperado de GitHub: <https://a-b-street.github.io/docs/project/history/retrospective/index.html>  
A/B Street. (s. f.). *A/B Street GitHub repository*. Recuperado de <https://github.com/a-b-street/abstreet>
2. **CityFlow (control semafórico con RL)**  
Chen, L., et al. (2019). *CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario*. arXiv preprint arXiv:1905.05217
3. **Datos móviles y generación/validación de demanda**  
Zilske, M., & Nagel, K. (2014). *Activity-based travel demand generation using mobile phone data*. Technical Report, TU Berlin. Recuperado de <https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2014/14-06/zilsk-e-nagel-abmtrans14-15feb2014.pdf>
4. **OpenStreetMap (red vial)**  
OpenStreetMap. (s. f.). *Map features*. Recuperado de [https://wiki.openstreetmap.org/wiki/Map\\_features](https://wiki.openstreetmap.org/wiki/Map_features)
5. **GTFS (transporte público)**  
GTFS. (s. f.). *General Transit Feed Specification reference*. Recuperado de <https://gtfs.org/documentation/schedule/reference/>
6. **MATSim (actividad-basada, gran escala)**  
MATSim. (s. f.). *MATSim – Multi-Agent Transport Simulation*. Recuperado de <https://www.matsim.org/files/book/partOne-latest.pdf>  
IVT ETH Zürich. (s. f.). *MATSim research*. Recuperado de <https://www.ivt.ethz.ch/en/research/matsim.html>
7. **SUMO (simulación microscópica de tráfico)**  
SUMO Developer Group. (s. f.). *SUMO publications & documentation*. Recuperado de <https://sumo.dlr.de/docs/Publications.html>
8. **POLARIS (ABM de alta performance)**  
Argonne National Laboratory. (s. f.). *POLARIS documentation*. Recuperado de <https://vms.taps.anl.gov/tools/polaris/documentation/>  
Sahin, O., & Stinson (?). *Title unknown*. Recuperado de [https://www.metrans.org/assets/upload/sahin\\_stinson-0.pdf](https://www.metrans.org/assets/upload/sahin_stinson-0.pdf)
9. **TRANSIMS (simulación regional histórica)**  
U.S. Department of Energy. (s. f.). *TRANSIMS project overview*. Recuperado de <https://www.osti.gov/biblio/88648>  
(Publicación técnica) Recuperado de <https://digital.library.unt.edu/ark:/67531/metadc793198/>
10. **Ride-pooling multi-operador**  
Journal article (2022). *[Título del artículo sobre ride-pooling y masa crítica]*.

Recuperado de Frontiers in Future Transportation:  
<https://www.frontiersin.org/journals/future-transportation/articles/10.3389/ffutr.2022.915219/full>

11. **Impactos ambientales y energéticos: SUMO y emisiones**

Journal article (2022). *[Título del artículo sobre SUMO y emisiones]*. Recuperado de ScienceDirect:  
<https://www.sciencedirect.com/science/article/pii/S1361920922001821>

12. **Simulación multi-agente aplicada a movilidad y revisión**

Artículo de revisión (2022). *[Título de revisión de ABM en movilidad urbana]*. Recuperado de Hindawi:  
<https://www.hindawi.com/journals/complexity/2022/1252534>  
Artículo de ScienceDirect (2021). *[Título sobre ABM y movilidad urbana]*. Recuperado de  
<https://www.sciencedirect.com/science/article/pii/S2590198221001913>

13. **COVID-19 / COVID-Mobile — calibración con datos móviles**

PMC. (s. f.). *Use of mobile phone data for demand calibration*. Recuperado de  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4852637/>  
Frontiers. (2021). *Data assimilation techniques in transport ABM*. Recuperado de  
<https://www.frontiersin.org/journals/future-transportation/articles/10.3389/ffutr.2021.660929/full>

14. **Barcelona – tarifa de congestión (peaje urbano)**

Expertos. (2018). *Simulación MATSim para tarifación urbana en Barcelona*. Recuperado de arXiv: <https://arxiv.org/abs/1803.06375>