



# Tecnológico de Monterrey

## **Simulación de Tráfico en la Ciudad de México**

Sebastián González, A01029746

Ana Paula Katsuda, A01025303

Karla Mondragón, A01025108

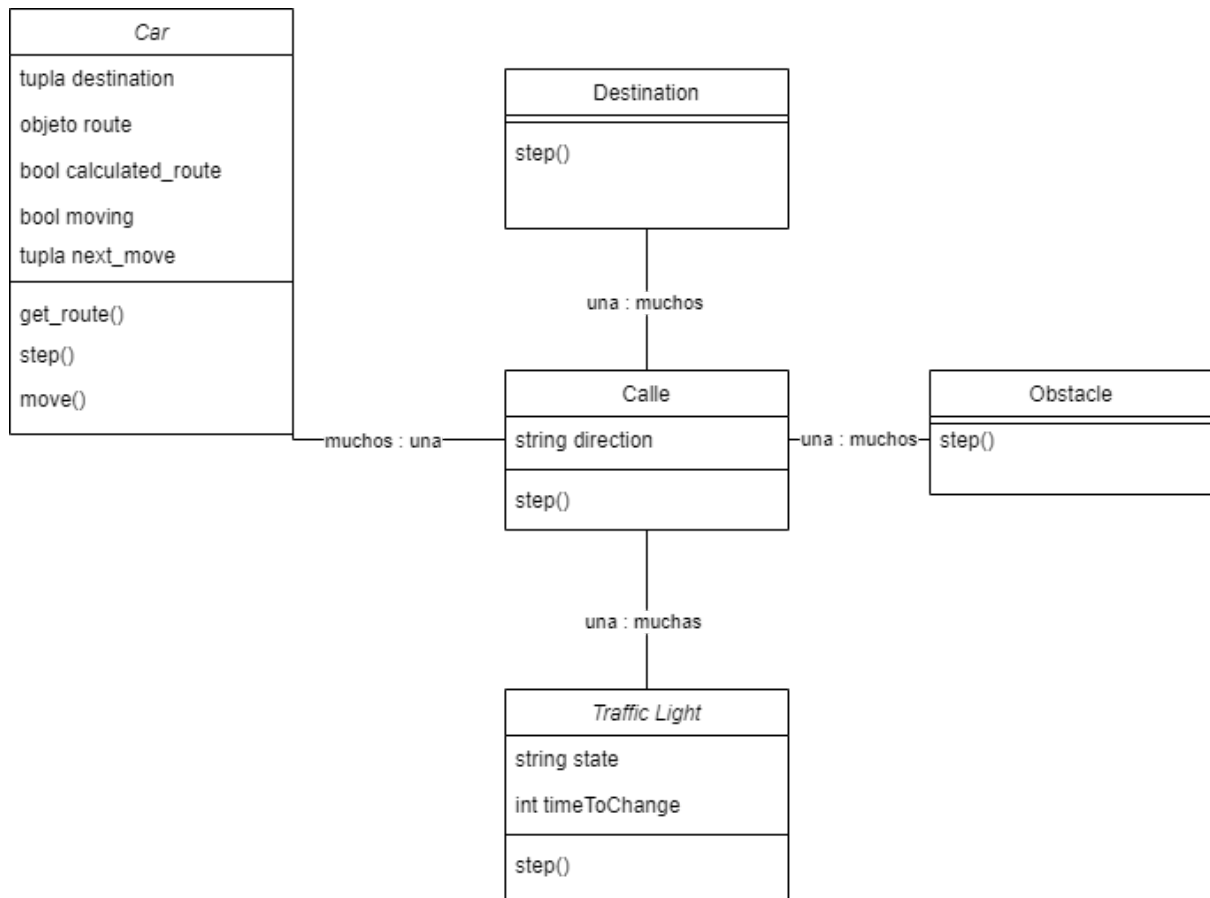
Andreína Sanáñez, A01024927

1 de diciembre 2022

# Índice

|                                  |          |
|----------------------------------|----------|
| <b>Índice</b>                    | <b>2</b> |
| <b>Diagrama de Clases</b>        | <b>3</b> |
| <b>Manual de Instalación</b>     | <b>4</b> |
| 1. Software                      | 4        |
| 1.1 Instalación de Python3.8     | 4        |
| 1.2 Instalación de Unity         | 4        |
| 1.3 Instalación de Git           | 4        |
| 2. Repositorio y Dependencias    | 5        |
| 2.1 Clonar Repositorio de Github | 5        |
| 2.2 Creación de Ambiente Virtual | 5        |
| 2.2 Instalación de Mesa          | 6        |
| 2.3 Instalación de Flask         | 6        |
| <b>Manual de Ejecución</b>       | <b>7</b> |
| 1. Correr Servidor               | 7        |
| 2. Abrir Proyecto en Unity       | 7        |
| 3. Correr Simulación             | 7        |
| <b>Video</b>                     | <b>8</b> |
| <b>Reflexiones Individuales</b>  | <b>8</b> |

## Diagrama de Clases



Link: [RetoCoches.drawio](https://retocoches.drawio.com/)

# Manual de Instalación

## 1. Software

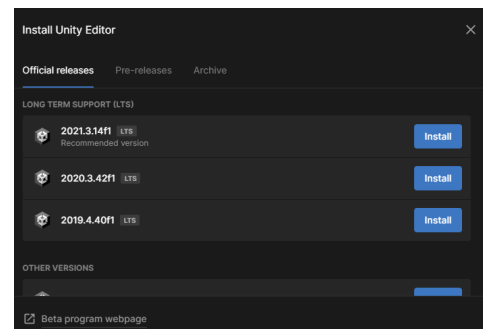
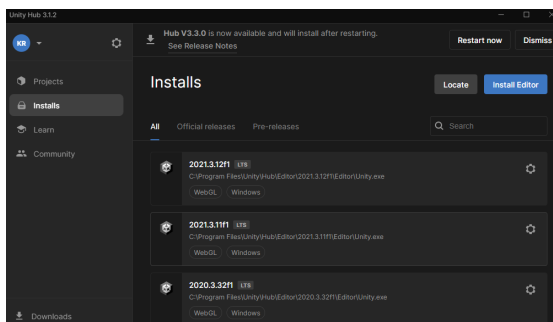
### 1.1. Instalación de Python3.8

Ingresa a la liga [Download Python](#) y selecciona la versión 3.8. Sigue las instrucciones del instalador.

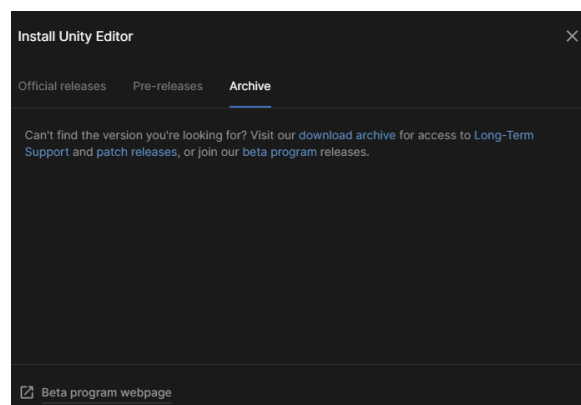
### 1.2. Instalación de Unity

Ingresa a la liga [Download - Unity](#) y selecciona la opción de “Download Unity Hub” con el sistema operativo correspondiente y sigue las instrucciones del instalador.

Una vez que se haya descargado Unity Hub, en la barra lateral selecciona la opción “Installs” y selecciona “Install Editor”.



Si no se encuentra dentro de las opciones la versión 2021.3.12f1, ingresa a la opción de “Archive” seguir el link de “download archive” y busca la versión dentro del apartado del 2021.



### 1.3. Instalación de Git

Ingresa a la liga [1.5 Getting Started - Installing Git](#) y selecciona el sistema operativo correspondiente. Sigue las instrucciones del instalador.

## 2. Repositorio y Dependencias

### 2.1. Clonar Repositorio de Github

Antes que nada, el repositorio de la presente aplicación se encuentra en el siguiente link:

[https://github.com/sebasgonvitec/TC2008B\\_Equipo3](https://github.com/sebasgonvitec/TC2008B_Equipo3)

Dentro de la barra de búsqueda del dispositivo ingresar “Git Bash”. Utilicé los comandos para moverse entre directorios y elegir donde se clonara el repositorio.

Ejemplo:

```
user@User MINGW64 ~ (main)
$ cd Documents
```

En Git Bash podemos escribir comandos después del signo “\$”.

El comando “cd” significa change directory y sirve para cambiar de directorio (carpeta). Para más información acerca de este tipo de comandos, diríjase a [CMD: 11 basic commands you should know \(cd, dir, mkdir, etc.\)](#)

Una vez situado en la carpeta en la que quiere clonar el repositorio, copia el siguiente comando:

```
git clone https://github.com/sebasgonvitec/TC2008B_Equipo3.git
```

Este comando crea una copia del repositorio dentro del dispositivo.

Cuando finalice la instalación del repositorio aparecerá dentro de Git Bash algo parecido a esto:

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
> Cloning into `YOUR-DIRECTORY`...
> remote: Counting objects: 10, done.
> remote: Compressing objects: 100% (8/8), done.
> remote: Total 10 (delta 1), reused 10 (delta 1)
> Unpacking objects: 100% (10/10), done.
```

### 2.2. Creación de Ambiente Virtual

Posicionarse en la carpeta del repositorio y dependiendo del sistema operativo en la terminal ejecutar los siguientes comando:

Para Windows:

```
> python3.8 -m agents venv  
> agents\Scripts\activate
```

Para Mac:

```
$ python3 -m agents venv  
$ . venv/bin/activate
```

### 2.3. Instalación de Mesa

Dentro del ambiente virtual, en la terminal ejecutar el siguiente comando:

```
$ pip install mesa
```

### 2.4. Instalación de Flask

Dentro del ambiente virtual, en la terminal ejecutar el siguiente comando:

```
$ pip install Flask
```

# Manual de Ejecución

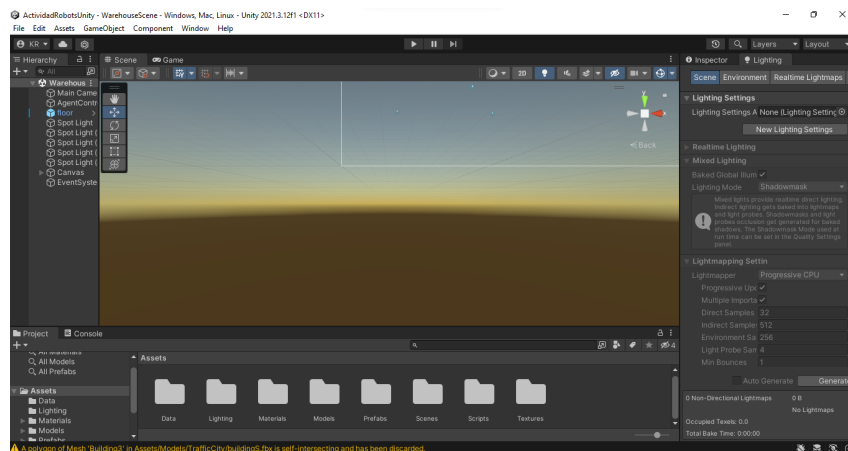
## 1. Correr Servidor

Posicionarse en la carpeta “past\_version” y ejecutar el siguiente comando:

```
TC2008B_Equipo3/reto/Server/past_version$ python3 server.py
```

## 2. Abrir en Unity

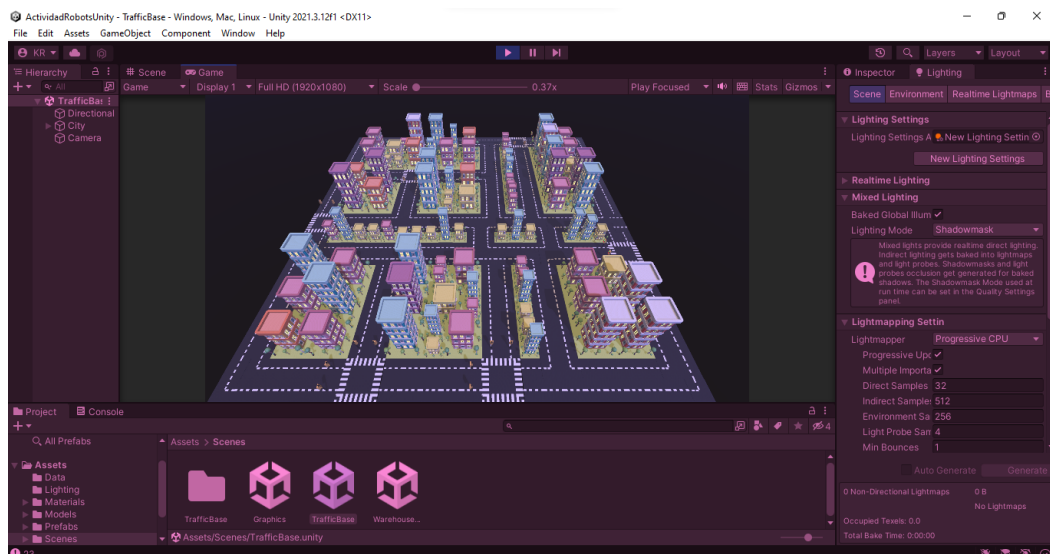
Abrir Unity Hub y para abrir el proyecto seleccionar la carpeta “ActividadRobotsUnity”.



Al llegar a la pantalla anterior, abrir la carpeta de escenas y seleccionar “Traffic Base”.

## 3. Correr Simulación

Dar click en play.



## *Video*

Link: [manual.mp4](#)

## *Reflexiones Individuales*

### A. Ana Paula

En cuanto a la solución desarrollada, sin duda alguna quedaron múltiples áreas de oportunidad. En primer lugar, se tenían distintas opciones de grafo y algoritmo de búsqueda que pudieron ser de gran utilidad. Se decidió utilizar un grafo dirigido cuyos nodos son las celdas del “grid” y el algoritmo BFS para buscar la mejor ruta de cada coche (dependiendo de su posición y del destino al que quiere llegar). En su momento, se tomó la decisión de utilizar esta estrategia para tener una mayor facilidad en el cálculo de la ruta y el grafo en sí. Las consideraciones más importantes que se tuvieron fueron las direcciones de las calles, el manejo de dichas direcciones en las intersecciones y la existencia de semáforos. Lo anterior con la finalidad de definir la manera en la que el coche se debe mover (debe interactuar) respecto a los otros elementos (en qué dirección debe avanzar, hacia qué calles tiene permitido entrar, si no choca con otros coches).

Algunas de las restricciones relevantes que se agregó al comportamiento de la simulación, fueron: no permitir que calles adyacentes se dirijan hacia la calle actual (relativo al eje de la calle actual), no permitir que se agreguen semáforos en diagonal (estos desembocan de otras calles en la intersección) y reaccionar al comportamiento tanto de los coches como del semáforo. De manera gráfica, se decidió hacer un diseño “low poly” y muy simple. Se eligió hacer esto puesto a que dichos diseños son muy estéticos y poco complejos, por lo que la simulación tendría la posibilidad de ser ligera (no alentarse mucho debido al peso de los diseños) manteniendo su atractivo visual.

Ahondando en el análisis de la solución presentada, se puede considerar útil al evitar los obstáculos desde el inicio, al calcular la ruta óptima en cuanto se crea el coche, al permitir reactividad por parte del coche en su entorno, y al considerar las direcciones de la calle. Si bien se tuvieron varios aciertos, hicieron falta distintas implementaciones y se pudieron utilizar mejores estrategias para la resolución.

Las principales áreas de oportunidad en este caso, son las siguientes: permitir una reactividad al tráfico más inteligente respecto al entorno del coche (revisar varias posiciones adelante para decidir si cambiar de carril o incluso cambiar de ruta), modificaciones útiles a la notación del mapa para evitar condiciones muy específicas (por ejemplo, puesto a que los semáforos no tenían dirección, se tuvieron que considerar en condiciones adicionales) y el uso de un grafo cuyos nodos sean las intersecciones en lugar de cada celda (con esta implementación se utilizarían pesos que consistirían en la distancia y cantidad de coches en entre las intersecciones).

En mi opinión, nuestra mayor área de oportunidad fue la creación de condiciones más generales. Creo que uno de los factores más influyentes en la falta de tiempo para la implementación fue que como equipo nos centramos en casos muy específicos sin tener el comportamiento básico de nuestros modelos. Pensando en mis expectativas al inicio del



bloque, puedo considerar que se cumplieron: efectivamente logré aprender sobre gráficos, agentes y sus comportamientos y la integración de estos. Tanto los conocimientos en clase como los errores que se fueron presentando durante la implementación influyeron en que desarrolle un gran aprendizaje en el tema. También, definitivamente comprendí las complejidades que puede tener la modelación de casos acercados a la realidad.

## B. Andreína

Antes que nada, es relevante comenzar diciendo que con la solución propuesta e implementada fue posible modelar de una forma considerablemente acertada el comportamiento de automóviles con objetivo de llegar a su destino a su vez que interactúan su ambiente, en este caso una ciudad. Esto gracias al modelo implementado donde se construyó un grafo dirigido en base al mapa dado (siendo cada celda del “grid” un nodo) y se utilizó el algoritmo de búsqueda BFS (*Breadth First Search*) para lograr encontrar la mejor ruta de cada carro partiendo de un punto de inicio y un destino. La principal ventaja y razón por la cual se utilizó este algoritmo fue debido a su facilidad para el cálculo de la ruta donde esté únicamente toma en consideración las distancias que implica el recorrido de cierto camino. Igualmente, al únicamente considerar distancias y hacer uso de un queue este resulta teniendo una mayor eficiencia a comparación de otros algoritmos como A\* que toman en consideración otros parámetros como el peso entre nodos, que en este caso podría representar el tráfico asociado con un camino.

Sin embargo, es importante mencionar que esta solución también cuenta con ciertas áreas de oportunidad, donde la más relevante de ellas está relacionada con la manera en la que se maneja la reactividad y comportamiento de los autos al enfrentarse al tráfico. Esto, ya que en el modelo se trata de reducir el tráfico a través de la consideración de un cambio de carril cuando el movimiento de cierto auto se ve obstruido por otro auto detenido enfrente del anterior. Aunque este método logra reducir la cantidad de tráfico al permitir el uso de todos los carriles en el mapa, su implementación en ocasiones puede ocasionar comportamientos no deseados al depender del estado de movimiento del auto frontal. Dicho esto una posible mejora podría ser el hecho de incorporar la cantidad de tráfico desde la definición del grafo, donde más aún se podría lograr contabilizar un alcance del tráfico que abarca toda la ruta, y no únicamente el coche de enfrente.

Por otro lado, otro aspecto de mejora que quizá vale la pena mencionar, es toda la parte de “inteligencia” que se encuentra en el modelo, donde a pesar de que en la medida de lo posible se establecieron reglas generales para definir el comportamiento del sistema hubo ciertos casos donde fue necesaria la especificación de casos particulares. En lo personal esto fue algo que realmente me llamó mucho la atención, ya que es en estas situaciones donde es notable lo complejo que puede llegar a ser la modelación computacional de ciertos fenómenos cuando esta se hace de forma explícita, y por ende lo útil que puede llegar a ser la exploración de otras áreas de la computación como Machine Learning para ofrecer soluciones lo más cercanas posibles a la realidad.

### C. Karla

El algoritmo que se decidió implementar fue BFS (Breadth First Search) esto debido al grafo que se implementó y que no se tenían peso entre los nodos. El grafo que se desarrolló consideró todas las celdas como nodos y es por esto que el peso (la distancia) entre los nodos siempre fue 1. Las ventajas de implementar este algoritmo, en vez de otros como A\* (que si se intentó) fue la velocidad con la que los carros avanzaban y que como mencionado anteriormente, no consideraba los pesos entre los nodos. Debido a múltiples factores como el tiempo de la materia y a las diversas ideas que se tenían dentro del equipo, dentro del proyecto consideramos que existen áreas de oportunidad importantes. Las mayores áreas de oportunidad que encontramos consisten que no incluimos una definición de tráfico, por lo que a pesar de que la simulación si modelaba carros en movimiento que eventualmente se convertían en tráfico, no se considero una toma de decisiones para los agentes si se encontraban dentro del tráfico; otra de las mayores áreas de oportunidades fue que no se recalculaba la ruta en ningún momento por lo que si el carro se encontraba dentro del tráfico su única opción era seguir en el carril o cambiarse de carril, pero nunca encontrar otra ruta.

Con respecto a mi aprendizaje y expectativas de la materia, definitivamente fue una de mis clases favoritas. Aprender acerca de sistemas multi-agentes que retratan la realidad y la inclusión de aspectos gráficos y de modelación 3D fue sumamente recompensante para mi y algo en que realmente quería ganar práctica. Al final, creo que si fuera una materia con más tiempo se habrían alcanzado mejores resultados y me hubiera gustado aprender más de ella.

### D. Sebastián

El desarrollo de este proyecto fue irónicamente una calle llena de baches. La primera parte fue decidir cómo abstraer el modelo de forma que los agentes pudieran entender. Se pensó primero en hacer el comportamiento completamente reactivo, de manera que el coche avanzará hacia su destino sin una ruta establecida. Al implementar esto nos dimos cuenta de que la lógica de seguir el sentido de las calles permitía que existieran ciclos no deseados y que por lo tanto nunca se llegará a un destino. Esta idea fue descartada.

La siguiente fue calcular la ruta desde un inicio y seguirla celda por celda hasta llegar al destino. El cálculo de esta ruta sería con algún algoritmo de pathfinding como lo es a estrella y Djikstra. La implementación de esta solución funcionó, pero el algoritmo de cálculo de ruta era muy lento y causaba un efecto de pausa cada vez que un agente nuevo aparecía en la escena. Usando el mismo grafo para la representación de los caminos y sus sentidos implementamos BFS. Si bien es cierto que este no es un algoritmo de pathfinding per se, según nuestra investigación iba a ser eficiente para encontrar la ruta en este caso.

Al final nuestra solución utiliza BFS junto con un grafo armado a la medida en base al mapa que se nos dió.

El proceso para evaluar diferentes opciones de algoritmos, diferentes soluciones y sobre todo el de desarrollar una solución solamente para darse cuenta de que no es la más óptima o no sirve, fue una experiencia muy valiosa y que definitivamente se suma a mis habilidades de solución de problemas.