

UNIVERSIDAD NACIONAL DE INGENIERÍA  
ESCUELA DE CIENCIA DE LA COMPUTACIÓN



**UNIVERSIDAD  
NACIONAL DE  
INGENIERÍA**

**Optimización de costos de viajes compartidos bajo  
demanda basados en aprendizaje por refuerzo**

Seminario de tesis:

Ciencia de la Computación

AUTOR

**Brando Miguel Palacios Mogollon**

Asesor

**César Jesús Lara Avila**

LIMA-PERU

2022

# Agradecimientos

Dedicado a mi padre y a mi madre por su apoyo incondicional a lo largo de la carrera.

A mi asesor César Lara Avila por su guía y compromiso para el desarrollo de la presente tesis.

A la universidad y a todos mis docentes, quienes me convirtieron en un profesional preparado para asumir todo tipo de desafíos.



# Resumen

Día a día, el tráfico vehicular y sus consecuencias traen consigo diversos factores negativos que afectan a la ciudades. Bajo estas circunstancias, los usuarios buscan nuevas modalidades de servicios de transporte que disminuyan el gasto de tiempo y dinero, junto con la contribución de una urbanización más sostenible. El viaje compartido surgido como solución potencial en la reducción de precios y duración del viaje, donde los pasajeros que poseen cierto interés según su condiciones y conveniencias puedan tener un viaje que sea lo más corto y económico posible es una alternativa.

En este proyecto se hace uso de grafos dirigidos no planares para representar toda una red de puntos de recojo y destino de distintos pasajeros que permita encontrar la ruta más inteligente para todas las solicitudes, incluyendo la posición del vehículo más cercano utilizando el conjunto de datos públicos de Chicago Taxi Trips. Utilizamos estos datos, para poder predecir un costo de esta ruta para cada pasajero buscando que este sea lo más económico y corto posible. Para esto, usamos técnicas de aprendizaje por refuerzo para asignar el vehículo más cercano a la ruta óptima de los pasajeros, usando el algoritmo Dueling DQN y el precio óptimo para el pasajero.

**Palabras claves:** viajes compartidos, aprendizaje por refuerzo profundo, transporte, grafos, precio, asignación, Dueling DQN

# Abstract

Every day, vehicular traffic and its consequences bring with it several negative factors that affect the cities. Under these circumstances, users are looking for new modes of transportation services that decrease the expenditure of time and money, along with the contribution to a more sustainable urbanization. Ridesharing emerged as a potential solution in reducing prices and travel time, where passengers who have a certain interest according to their conditions and convenience can have a trip that is as short and economical as possible. This would not be possible without a good route and planning according to the available vehicles. In this project, non-planar directed graphs are used to represent a whole network of pick-up and drop-off points for different passengers in order to find the most intelligent route for all requests, including the position of the closest vehicle. Where finally, according to the public data set of Chicago Taxi Trips in the area, to predict a cost of this route for each passenger looking for this to be as economical and short as possible. For this, we use reinforcement learning techniques to assign the closest vehicle to the optimal passenger route, using the Dueling DQN algorithm connected to a neural network connected to the dataset, thus giving the optimal price for the passenger.

**Palabras claves:** ridesharing, deep reinforcement learning, transportation, graphs, pricing, assignment , Dueling DQN

# Índice general

<b>1. Introducción</b>	<b>10</b>
1.1. Generalidades . . . . .	10
1.2. Problemática . . . . .	14
1.3. Objetivos . . . . .	15
1.3.1. Objetivo general . . . . .	15
1.3.2. Objetivos específicos . . . . .	15
1.4. Alcance y limitaciones . . . . .	16
1.4.1. Hipótesis . . . . .	16
1.4.2. Alcance . . . . .	16
1.4.3. Limitaciones . . . . .	16
1.4.4. Aportes . . . . .	16
<b>2. Estado de arte</b>	<b>17</b>
2.1. Objetivo de la revisión . . . . .	17
2.2. Método usado en la revisión . . . . .	17
2.3. Estado del Arte . . . . .	18
<b>3. Marco teórico</b>	<b>22</b>
3.1. Problema de enrutamiento de vehículos (VRP) . . . . .	22
3.1.1. Variantes del VRP . . . . .	22
3.2. Viajes compartidos . . . . .	23

3.2.1.	Tipos de viajes compartidos . . . . .	24
3.2.2.	Precios . . . . .	25
3.2.3.	Patrones y variables . . . . .	25
3.2.4.	Complejidad . . . . .	28
3.2.5.	Beneficios y riesgos . . . . .	29
3.3.	Camino Hamiltoniano . . . . .	29
3.3.1.	Definición formal . . . . .	29
3.3.2.	Digrafo Hamiltoniano . . . . .	30
3.3.3.	Teorema de Ore y Dirac . . . . .	30
3.4.	Programación Entera y Lineal . . . . .	31
3.5.	Algoritmos Genéticos . . . . .	31
3.6.	Aprendizaje por refuerzo . . . . .	33
3.6.1.	Q-Learning . . . . .	34
<b>4.</b>	<b>Metodología</b>	<b>36</b>
4.1.	Base de datos . . . . .	36
4.1.1.	Conjunto de datos Chicago Taxi Trips . . . . .	36
4.1.2.	Boundaries Community Areas . . . . .	38
4.2.	Preprocesamiento . . . . .	39
4.3.	Tecnologías . . . . .	40
4.3.1.	Docker . . . . .	41
4.3.2.	PostgreSQL . . . . .	41
4.3.3.	OpenStreetMap . . . . .	42
4.3.4.	Librerías de Python . . . . .	42

4.4. Planteamiento . . . . .	43
4.4.1. Planteamiento matemático . . . . .	47
4.5. Definición del modelo matemático . . . . .	49
4.5.1. Variables de decisión . . . . .	49
4.5.2. Función objetivo . . . . .	49
4.5.3. Restricciones . . . . .	50
4.6. Definición del modelo genético . . . . .	50
4.6.1. Función Fitness . . . . .	51
4.6.2. Operadores . . . . .	51
4.7. Definición del modelo de aprendizaje por refuerzo . . . . .	52
4.7.1. Dueling DQN . . . . .	52
4.8. Estructura del modelo de aprendizaje por refuerzo . . . . .	53
4.9. Red multi-modelo de precios . . . . .	54
4.10. Flujo conceptual . . . . .	55
<b>5. Resultados y Discusiones</b>	<b>57</b>
5.1. Métricas . . . . .	57
5.2. Comparación por modelo . . . . .	57
5.3. Observaciones según el modelo . . . . .	60
5.4. Gráficas . . . . .	60
<b>6. Conclusiones</b>	<b>62</b>
6.1. Conclusiones . . . . .	62
6.2. Trabajos futuros . . . . .	63

# Índice de figuras

1.1.	Gráfica comparativa antes y después de la pandemia respecto a la forma de movilizarse, por Lima Cómo Vamos, 2021 . . . . .	11
1.2.	Tabla de valoración del servicio según el tipo de transporte, por Lima Cómo Vamos, 2022 . . . . .	12
1.3.	Gráfica de linea sobre las ganancias de los viajes compartidos desde el 2017 hasta abril del 2022, por Bloomberg Second Measure (Perri, 2022). . . . .	13
3.1.	Ilustración del patrón 1, viaje idéntico. . . . .	26
3.2.	Ilustración del patrón 2, viaje incluido. . . . .	26
3.3.	Ilustración del patrón 3, donde, (i). $u_b = pk_b$ , (ii). $v_b = dp_b$ . . . . .	27
3.4.	Ilustración del patrón 4, viaje por desvío. . . . .	27
3.5.	Ilustración de la interacción entorno-agente. . . . .	33
3.6.	Ilustración esquemática de la red profunda usada para el aprendizaje por refuerzo profundo del videojuego de ATARI. (Mnih et al., 2015) . . . . .	35
4.1.	Grafica del modelo de base de datos. . . . .	40
4.2.	Ilustración de mapa junto con datos geográficos de viajes. . . . .	44
4.3.	Ilustración de posibles rutas tomadas para el mismo problema de viajes . . . . .	45
4.4.	Ilustración de selección del mejor vehículo por cercanía a los puntos. . . . .	45
4.5.	Ilustración de puntos geoespaciales conectados. . . . .	46
4.6.	Ilustración de aristas según el tipo de vértices. . . . .	46
4.7.	Ilustración de grafo de red de recojo-destino extendido. . . . .	47

4.8. Ilustración comparativa entre una arquitectura DQN clásica y una Dueling DQN. (Wang et al., 2015) . . . . .	53
4.9. Ilustración de la estructura del modelo. . . . .	54
4.10. Ilustración de la estructura de la red neuronal. . . . .	55
4.11. Ilustración del flujo de la red neuronal de precios conectado a los modelos comparativos. . . . .	56
4.12. Ilustración del flujo de conexión del proyecto. . . . .	56
5.1. Gráfica de loss vs epoch por cada 10 iteraciones. . . . .	58
5.2. Gráfica comparativo entre la diferencia de precios. De rojo, Dueling-DQN vs la variante VRP usando LP y de verde, algoritmo genético canonico vs el Dueling-DQN	58
5.3. Barra comparativa de pruebas post aprendizaje. . . . .	59
5.4. Gráficas de los puntos geográficos según los caminos tomados en diferentes representaciones. . . . .	61

# Introducción

## 1.1. Generalidades

El CEPAL (Comisión Económica para América Latina y el Caribe), propuso a la movilidad sostenible como uno de los principales factores en la calidad de vida y medio ambiente, dándole lugar al uso de trasportes colectivos, como la mejor salida, observando la necesidad de una inversión en infraestructura y estrategia de financiamientos para mitigar la posible brecha causada a niveles sociales y económicos en América Latina. (CEPAL, 2022)

Es así que, el transporte urbano como los conocemos, toma lugar importante como el medio más común e importante en un sociedad que necesita movilizarse con mayor rapidez a sus destinos. Sin embargo, este movilización puede variar según una lista larga de condiciones que pueden afectarnos tanto de manera social como económica. Siendo de interés, como sociedad, lograr un equilibrio que nos permita agilizar nuestra forma de viajar.

Esta situación se ha visto afectada los 3 últimos años, tras la aparición de una nueva pandemia mundial, causada por el coronavirus **SARS-COV2**. Haciendo que el estado peruano declare un estado en emergencia y decrete cuarentena obligatoria mediante el **Decreto Supremo N° 008-2020-SA**, impidiendo el flujo de vehículos de forma indefinida. No obstante, con el tiempo se fue adaptando dependiendo de la condición nacional sobre este virus. (Merriam-Webster, s.f.; Ministerio de Salud, 2020)

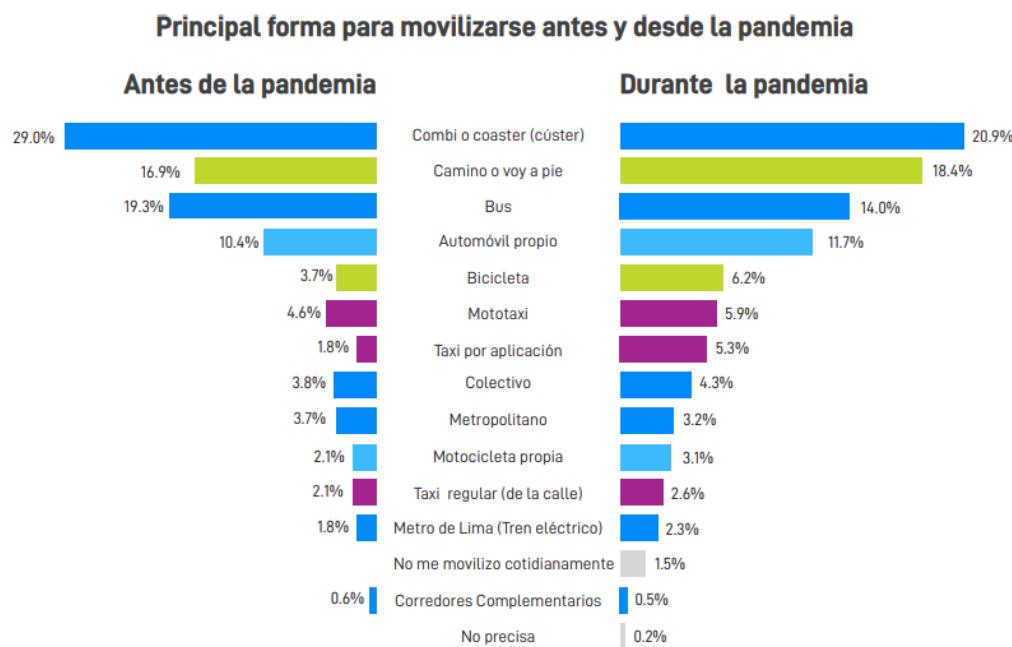
Hoy en día, con el **Decreto Supremo N° 130-2022-PCM** fue derogado el estado de emergencia de forma definitiva y con el toda restricción sanitaria, haciendo que el transporte retorne a su flujo esperado; sin embargo, este hito histórico marcó en las necesidades de los ciudadanos de todo el mundo, influyendo en sus decisiones ante necesidades primarias, entre ellas, el transporte. (Presidencia del Consejo de Ministros, 2022).

Para el Instituto Nacional de Estadística e Informática (INEI), en el 2019 el Perú tenía en circulación 3 millones de vehículos a nivel nacional, donde el 1 982 650 están en la capital (66 %). Esto debido a que, la capital posee alrededor de 10 millones de personas con la necesidad de desplazarse por diferentes partes de Lima. Este hecho, nos ha llevado a entender la sobre demanda de vehículos en la capital, siendo la flota de vehículos públicos y compartidos muy

pocos para la cantidad de personas en un la ciudad. (Instituto Nacional de Estadística e Informática, 2020)

La encuestadora “Lima Como Vamos” ha realizado diversas encuestas muy importantes que indican las necesidades de los pobladores, durante y después de la pandemia, demostrando este cambio de intereses de los ciudadanos de Lima y Callao. En el *Informe urbano de percepción ciudadana en Lima y Callao 2021* podemos ver la comparativa de la principal forma de movilizarse antes de la pandemia y después de la pandemia (Fig 1.1), donde vemos un incremento en el uso de vehículos privados, un incremento del 4 % en el uso de taxis, además de la desconfianza de la población en utilizar movilidades masivamente colectivas (buses, Alimentadores, Trenes).

Adicionalmente, en el informe realizado para el 2022, un cambio post pandemia, se destaca los puntos más valorados por medios de trasportes urbanos, destacando a los taxis por aplicación como el más seguro, puntual y cómodo de usar sobre los demás (Fig 1.2). Conjuntamente, a nivel medio ambiental, la acuesta apunta a que el mayor problema para los ciudadanos es la contaminación del aire por vehículos. (Lima Cómo Vamos, 2021, 2022).



**Figura 1.1:** Gráfica comparativa antes y después de la pandemia respecto a la forma de movilizarse, por Lima Cómo Vamos, 2021.

Bajo este contexto, muchas empresas a nivel mundial han luchado por modernizar sus negocios post pandemia manteniendo la oferta sobre la demanda del país, y las compañías de transporte privado no son la excepción. Para esto, las empresas también denominadas *empresas de redes*

	TAXI POR APLICACIÓN	TAXI TRADICIONAL	METROPOLITANO	METRO DE LIMA	CORREDORES COMPLEMENTARIOS	BUSES, CUSTERS Y COMBIS
La rapidez	21.2%	27.1%	49.8%	63.8%	33.4%	20.4%
La seguridad	20.9%	6.2%	7.7%	5.5%	7.6%	4.1%
La puntualidad	16.6%	5.6%	9.2%	7.6%	8.7%	2.2%
La comodidad	12.3%	11.0%	5.6%	4.6%	10.3%	4.4%
El costo	12.0%	22.7%	5.5%	7.3%	11.4%	24.3%
El trato del personal	7.8%	9.4%	1.5%	2.1%	1.9%	4.1%
La limpieza	4.9%	7.2%	8.5%	4.7%	9.1%	2.8%
La cobertura de la ruta	1.6%	5.1%	7.9%	1.3%	5.1%	20.1%
El orden	0.9%	1.0%	1.7%	1.7%	6.9%	2.2%
No sabe	1.8%	4.6%	2.7%	1.4%	5.8%	15.5%

Figura 1.2: Tabla de valoración del servicio según el tipo de transporte, por Lima Cómo Vamos, 2022.

de transporte (Transportation Network Company, TNC), las que permiten la conexión entre conductores y pasajeros bajo plataformas digitales, se han fijado en el transporte urbano del país tales como: Uber, Cabify, InDriver, DiDi, entre otros. Esta creciente demanda de medios de transporte ha provocado el aumento de tráfico en la capital. (The Nexus, 2021)

Es importante mencionar que las TNC han cambiado su modelo de negocio para adaptarse a esta nueva era post pandemia, buscando que sus beneficios sean cada vez mejores, como muestra de ello, **Uber Technologies, Inc.**, una de las empresas destacadas en el mercado de TNC, sufrió una caída grande desde el inicio de la pandemia viviendo su peor época en abril del 2020, donde su demanda bajo hasta un 80 % respecto al 2019. Tras esta pérdida, hubieron cortes masivos de personal y oficinas. Tras este golpe, Uber realizó una reforma en sus servicios. Entre ellos, **Uber pool**, ya que el servicio se basaba en compartir el transporte con personas cercanas para aminorar la tarifa de viaje, donde el distanciamiento social y las leyes de seguridad sanitaria a causa de la crisis pandémica, hicieron que este sea servicio ilegal y seguro durante la pandemia, el cual tuvo que ser cancelado hasta recién a mediados del 2021. ((Bond, 2020; Rana, 2020))

Sin embargo, esta caída no desacreditó sus beneficios, lo que llevó a reinventarse como **UberX share**, afirmando que el servicio da un ahorro de hasta el 30 % del viaje únicamente para 2 pasajeros usando cubre bocas de manera obligatoria. Entre las condiciones de Uberx share está el horario de servicio, estando disponible bajo un horario fijo. Actualmente solo algunos países

ya poseen este servicio, que tiene como objetivo potenciar el mercado de la empresa, en razón de que, según The Wall Street Journal, no ha tenido buenos resultados en la post pandemia, dando una baja en la bolsa del -44.12 % respecto a los últimos 3 años. (Ambito, 2022; The Wall Street Journal, 2022; Uber Blog, 2022).

Desde otro ángulo, en ciudades como Beijing encontramos un gran desafío por la densidad demográfica que se presentan en las zonas más urbanizadas; para solucionar este problema la compañía DiDi Chuxing, evaluó la incorporación de lugares de recojo y despacho más explícitos y fáciles con las descripciones de los pasajeros de su posición, con el objetivo de un ahorro de energía. Según el noticiero chino, SohuNews (2017), el servicio titulado “Station Carpool” logró reducir en un 30 % el kilometraje consumido por los automóviles. Método que tomó Uber para la creación de su servicio “Express Pool” (Uber Blog, 2018) y Lyft con “Shader Saver” (Lyft, 2019). Además, en Estados Unidos según Bloomberg Second Measure, las ventas de los viajes compartidos han aumentando gradualmente post pandemia (UberX Share 80 %, Shader Saver 49 %). Por lo anterior, según la Figura 1.3, Uber sigue dominando este mercado conteniendo el 72 % de las transacciones de viajes compartidos. Con esto nos podemos dar cuenta la participación del negocio de los viajes compartidos sigue siendo un servicio útil y necesario para la comunidad internacional. (Perri, 2022).

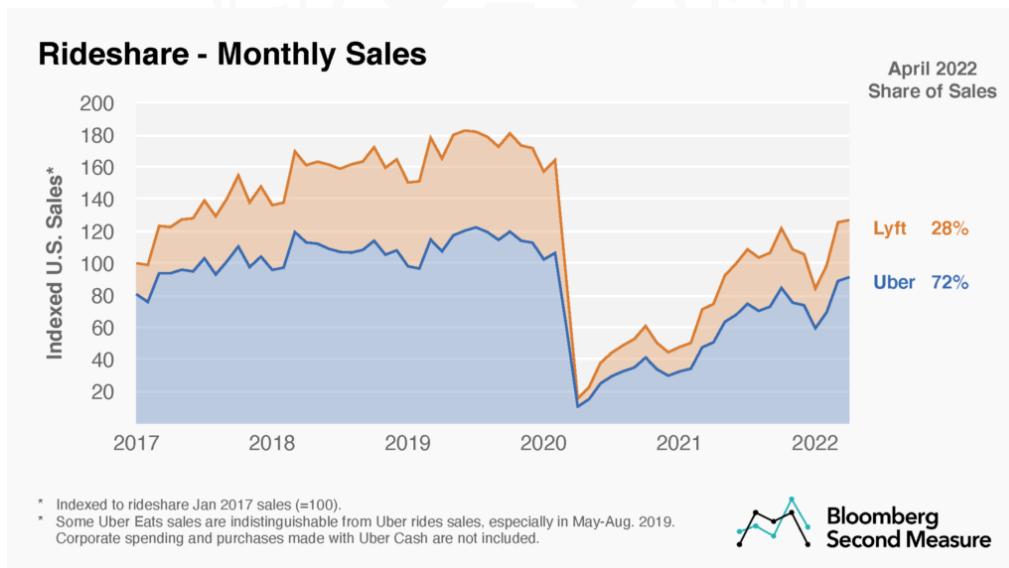


Figura 1.3: Gráfica de línea sobre las ganancias de los viajes compartidos desde el 2017 hasta abril del 2022, por Bloomberg Second Measure (Perri, 2022).

## 1.2. Problemática

El transporte del día a día, es actualmente, una espada de doble filo. Mientras en un lado, buscamos vehículos privados que nos den comodidad y ahorrarnos el tiempo de espera que los trasportes públicos no nos dan, en el otro lado, el excesivo uso de vehículos aumenta el tráfico vehicular en la ciudad retrasándonos aun más de lo que esperábamos evitar.

El planteamiento de creación de nuevas carreteras, vías alternas, autopsias, entre muchas otras, podría ser una buena idea, no obstante, sigue siendo la menos efectiva en cuanto a inversión, y el aumento constante de los vehículos en la ciudad. Además, los servicios de redes de transporte son muy efectivos en cuanto a comodidad y seguridad hablamos, sin embargo, no es nada económico realizar viajes recurrentes a largo plazo.

En este contexto se describe la pregunta que motiva la presente investigación:

**¿Cómo podemos disminuir el kilometraje de los viajes a vehículos que realizan transporte privado y beneficiar a los pasajeros?**

Para este fin, buscamos tener un modelo de transporte que logre usar menos kilometraje para sus viajes, para ahorrar tiempo y usando menos combustible. Entre muchos modelos de viajes, podemos destacar al servicio de viajes compartidos, siendo uno de los servicios privados de gran demanda internacional para las TNCs, por su eficiencia, ajuste económico y ayuda medioambiental.

El servicio de viaje compartido, del inglés *ridesharing*, permite a los viajeros compartir vehículos con aquellos que tienen orígenes, destinos y horarios similares con poca antelación, lo que reduce el número de vehículos en circulación en una área al aumentar el número de pasajeros por transporte. Este proceso permite a los pasajeros compartir el costo del conductor de forma justa y proporcionada a sus destinos. (Bhogal, 2022)

Sin embargo queda otra problemática pendiente:

**¿Como el campo de la informática será de beneficio en este problema?**

Para poder realizar este tipo de servicios requerimos con una considerable cantidad de variables necesarias para estimar una distancia y precio a pagar al conductor, como lo son: los puntos de recojo de cada pedido, los vehículos más cercanos a los pasajeros, el precio por kilómetro, entre otros. Es aquí que nos vemos beneficiados por una rama de la ciencia que brinda soluciones a grandes problemáticas imitando el comportamiento e inteligencia humana: la Inteligencia Artificial, en específico el aprendizaje por refuerzo; destacando por su versatilidad para con este

tipo de problemas, adaptándose a las situaciones que se le muestre.

Durante este proceso de debemos tener bien en claro los conceptos que implican en un viaje compartido, ya que, no solo se puede depender de que pasajeros son aptos para tener una ruta única que puedan viajar, sino también, en que posición se encontrará el automóvil, siendo indispensable de reconocer para determinar una ruta óptima y que se encuentre en el momento y lugar óptimo para tomar el pedido.

Para lograr dicho propósito, se busca utilizar diferentes tecnologías y definiciones matemáticas que describan la solución al problema, como la construcción de un grafo dirigido para representar de forma zonal partes de un mapa, un modelo de aprendizaje por refuerzo profundo que interactúe con el grafo dirigido, una red neuronal que use las variables para estimar el precio del pedido, etc.

En resumen, el servicio de viajes compartidas puede llegar a ser una respuesta al problema del tráfico vehicular, facilitando el desplazamiento de personas en un mismo vehículo disminuyendo el consumo de combustibles y el tiempo de espera, es por ello que con esta investigación, buscamos sentar las bases a soluciones futuras ya sea para seguir incentivando el aprendizaje por refuerzo aplicado al transporte y para servir como base a futuras investigaciones.

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Desarrollar una propuesta de optimización al problema de viajes compartidos aplicados para el cálculo de precios basado en las técnicas de aprendizaje por refuerzo profundo (DRL)

### **1.3.2. Objetivos específicos**

1. Identificar la importancia de la optimización de precios en los servicios de transporte compartido.
2. Formular algoritmos de solución basada en técnicas de inteligencia artificial con los parámetros del entorno estudiado que afectan la arquitectura propuesta.
3. Diseñar una arquitectura de solución que permita minimizar los precios de viajes compartidos respecto a pedidos de viajes privados.
4. Evaluar otras propuestas de solución adaptadas a los viajes compartidos.

## 1.4. Alcance y limitaciones

### 1.4.1. Hipótesis

Mediante el uso de grafos y modelos de aprendizaje por refuerzo, es posible representar pedidos de viajes compartidos y carreteras que permitan determinar que pedidos podemos seleccionar en una lista de pedidos pendientes haciendo posible que ambos compartan un mismo vehículo según las condiciones de la ruta, disminuyendo el costo del viaje y el kilometraje.

### 1.4.2. Alcance

El proyecto se centra en el procesamiento de datos de una ciudad y como esta maneja sus pedidos de taxis. Por lo tanto, se enfoca en el área de transporte urbano y como encontrar un control en el transporte moderno ,y por consiguiente, el tráfico vehicular. Con esto buscamos obtener un modelo que se adapte a cada ciudad que se le presente y tenga los parámetros necesarios para estimar precios reales en sus calles.

### 1.4.3. Limitaciones

El proyecto usa información publica del manejo de taxis en una ciudad, para su entrenamiento y toma de decisiones en la practica, en caso la ciudad no tenga un conjunto de datos requerido, las estimaciones como el precio y kilometro, fallen; es por eso que no toda ciudad del mundo puede calificar para tener este tipo de soluciones, ya que carecen de datos importantes para la misma. Además, los modelos estudiados requirieron de un coste computacional grande, por lo que los resultados demoran más de lo estimado. Asimismo, las investigaciones en el país sobre el tema en específico son pocas.

### 1.4.4. Aportes

La técnica planteada para la estimación de costos en los viajes compartidos de taxis, basados en métodos de Inteligencia Artificial, es un aporte al campo de la Optimización como una variante del problema de enrutamiento de vehículos y plantea las bases a futuras aplicaciones al problema de los viajes compartidos, sabiendo que este tipo de servicio, trae consigo beneficios económicos y medio ambientales para la sociedad.

# Estado de arte

En este capítulo se hace la revisión de las investigaciones previas donde se están utilizando técnicas de aprendizaje por refuerzo y otros algoritmos relacionados al enrutamiento, asignación y estimación de precios de rutas para servicios de transporte compartido. Se realizó la búsqueda de los estudios previos mediante cadenas de búsqueda, para encontrar información útil para posibles estrategias de solución.

## 2.1. Objetivo de la revisión

El objetivo de esta búsqueda es encontrar estrategias del planteamiento del problema y herramientas para la implementación y fundamentos de la problemática planteada para la investigación.

## 2.2. Método usado en la revisión

Se planteó el uso de preguntas iniciales que impliquen los objetivos y además cadenas de búsqueda en las bases de datos Scopus, IEEEExplore, ArXiv y ACM Digital Library. Al tratarse de un tema de aplicación novedoso y que aun posee continuidad se buscó las revisiones más recientes posibles.

Usamos un criterio de inclusión y exclusión para nuestra revisión, donde se realizó un proceso de filtrado de bibliografía útil para nuestro objetivo. Por esto mismo, de todos los documentos recopilados incluimos los que tenían una arquitectura de aprendizaje por refuerzo profundo, uso de grafos, fijación de precios y algoritmo de búsqueda que solucione la complejidad de la estimación de precios en viajes compartidos; se excluyó, los documentos relacionados a aprendizaje por refuerzo multi-agente, soluciones planteadas sobre topología de redes de computadoras y variantes de aprendizaje por refuerzo profundo que requieren altos recursos de computadora, considerando que se aleja del planteamiento del uso de una aprendizaje por refuerzo profundo y del impacto sobre nuestro enfoque de aplicación.

En la Tabla 2.1 se en listan las bases de datos revisadas y la consulta realizada en cada una.

Base de Datos	Consulta	Resultados
Scopus	TITLE-ABS-KEY ( reinforcement learning AND (pricing OR price) AND (ridesharing OR ride sharing) AND graph AND (LIMIT-TO ( PUBYEAR , 2022 ) OR LIMIT-TO ( PUBYEAR , 2019 ))	Article (34) Books (5)
ACM	"query": {AllField:(reinforcement learning) AND AllField:(transportation AND pricing OR price) AND AllField:(ride sharing OR ridesharing) AND AllField:(NOT (robot AND multi agent)) AND AllField:(routing OR transportation OR traffic)} "filter": {ACM Content: DL}	Article (21) Books (3)
IEEEExplore	(All Metadata":reinforcement learning) OR (All Metadata":ridesharing OR All Metadata":ridesharing ) AND (All Metadata":transportation OR All Metadata":taxi) AND (All Metadata":routing) NOT (All Metadata":robot AND All Metadata":multi)	Article (43)

Cuadro 2.1: Cuadro de consultas en Bases de Datos Bibliográficas.

### 2.3. Estado del Arte

Realizando el análisis exhaustivo de todas las revisiones con respecto a palabras específicas se obtuvieron 28 fuentes. Donde se tomaron en cuenta las 9 más relevantes.

- *A Distributed Model-Free Ride-Sharing Approach for Joint Matching, Pricing, and Dispatching Using Deep Reinforcement Learning*, modelo completo de fijacion de precios planteando la relación driver-passenger y los parámetros que involucra la ruta, como lo son las ventanas de tiempo, el cantidad de pasajeros compartidos, la distancia recorrida, el tipo de vehículo, entre otros; usando las capacidades de cada vehículo genera rutas óptimas con un algoritmo de emparejamiento cruzado usando un modelo distribuido de DQN. (Haliem et al., 2021)
- *An Application of Reinforced Learning-Based Dynamic Pricing for Improvement of Ridesharing Platform Service in Seoul*, el estudio se realizó para resolver el problema en el peor periodo

de tiempo aplicando tasas de aumento según las zonas con mayor y menor demanda en beneficio de la plataforma, lo que permite reducir en mayor medida los problemas de equidad regional relacionados con los servicios de viajes compartidos en Seúl. (Song et al., 2020)

- *Empty-Car Routing in Ridesharing Systems*, toma en cuenta un modelo de una red de colas basada en estados, el cual toma en cuenta carros completamente vacíos y los enruta con pedidos. Logran la optimización del flujo de automóviles mediante una optimización basada en límites, los resultados predicen el flujo de carros vacíos en horarios del día, llegando a un punto de equilibrio. (Braverman et al., 2019)
- *A Distributed Model-Free Ride-Sharing Approach for Joint Matching, Pricing, and Dispatching Using Deep Reinforcement Learning*, se desarrolla un servicio que se centra en generar rutas óptimas para cada vehículo, además el algoritmo planteado está basando en la recompensa donde el flujo esta influenciado y calculado por una Deep Q-network, que mediante una infraestructura de negocio, usa el DQN relacionado al estado del vehículo (Vehicle) y pasajero (Customer) mapeado por un OSRM (Open Source Routing Machine) para obtener la distancia de la ruta y la trayectoria posible de los vehículos. Se considera importante la demanda en línea, los precios asociados con cada viaje, las capacidades y ubicaciones del vehículo, para realizar sus decisiones. Este modelo fue entrenado por el conjunto de datos públicos de New York City Taxi. (Haliem et al., 2021).
- *Optimal Routing for Autonomous Taxis using Distributed Reinforcement Learning*, plantea resolver el problema en tiempo real, donde su arquitectura de basa en un proceso de decisión de Markov (MDP), planteando un algoritmo SARSA distribuido para capturar los errores del modelo y el cambio frecuente del entorno como la posición de clientes, tráfico y tarifas. Ellos aplican este modelo a todos sus agentes, donde el algoritmo SARSA realiza de tarea de asignación para designar a cada carro un cliente libre. (Rahili et al., 2018).
- *Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case*, se muestra una análisis significativo del problema de toma de decisiones mediante DRL el cual busca solucionar los problemas de optimización de enrutamiento, dado que una red neuronal presenta dificultades para predecir un entorno que dependa de una estructura de grafos, dado que tendrá problemas para solucionar redes neuronales completamente conectadas. En este documento se analiza la posibilidad de utilizar Graph Neural Networks (GNN) en conjunto con DRL. Donde GNN tiene la tarea de modelar los grafos propuestos y en conjunto tiene la capacidad de operar bajo una topología. El proyecto se analizó en un escenario de transporte óptico (OTN). (Almasan et al., 2020).

- *Anticipatory routing methods for an on-demand ridepooling mobility system*, para este documento se propone el uso simultáneo de los recursos, es decir vehículos. Tomando en cuenta el desequilibrio entre la demanda de estos recursos, respecto de la oferta circundante en la ciudad. La técnica anticipatoria asigna a usuarios enclaves previo a los otros pasajeros y con el tiempo modifica el estado de la ruta más eficiente para futuras solicitudes, aunque parezca una solución netamente suppositoria, el sistema se basa en las solicitudes actuales y las recientes para reconocer las zonas de alta demanda. Se propone la generación de datos artificiales para que el aprendizaje sea uniforme. Se destaca la forma de incluir movimiento de vehículos logra solucionar la tasa de rechazos bajo movilizándose continuamente a zonas más demandadas. (Fielbaum et al., 2021)
- *Trip-Vehicle Assignment Algorithms for Ride-Sharing*, se utiliza un grafo de  $n$  vehículos con el objetivo de asignar a un vehículo el peso mínimo. Se tomo como base un máximo de 2 solicitudes por vehículo. Y se realizó algoritmo de asignación dinámica que se basa en una solución del problema de cierre Minimum Wocho Mcon vértice Penalty (MWMP). (Li et al., 2020)
- *A Matching Algorithm for Dynamic Ridesharing*, un viaje por partido dinámico es la asignación automática de las solicitudes sin previa validación. Bajo este planteamiento, se usó un algoritmo de coincidencias que verifique si un nuevo pasajero puede entrar al vehículo sin lograr un umbral de desvió máximo, que ayuda a saber si generar esa nueva ruta será la mejor decisión. (Schreieck et al., 2016)
- *DeepPool: Distributed Model-free Algorithm for Ride-sharing using Deep Reinforcement Learning*, DeepPool se basa en una Deep Q Learning (DQN) para aprender políticas de despacho y recojo, reconoce la demanda de viajes por zonas más requeridas que otras. Se reconoció la rápida adaptabilidad del modelo para esta como para servicios no compartidos. (Al-Abbasi et al., 2019)
- *A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion*, planteamiento de un problema de asignación en base de proponer alternativas de ruta, para esto, utiliza una arquitectura que registra y opera dinámicamente los vehículos según un algoritmo de K- caminos mínimos dependientes del tiempo (Zhou y Roncoli, 2022)
- *Algorithms for Trip-Vehicle Assignment in Ride-Sharing*, en este documento se propone una solución algorítmica según el agrupamiento de automóviles, sabiendo que este problema planteado es NP-Hard, el modelo planteado genera un coste 2.5 veces mayor al óptimo como máximo. (Bei y Zhang, 2018)

- *The Optimization Model of Ride-Sharing Route for Ride Hailing Considering Both System Optimization and User Fairness*, optimización del problema de viajes compartidos mediante algoritmos genéticos, siendo nuestro objetivo obtener la distancia mínima de tomar nuestros viajeros, donde tenemos un algoritmo cerca al óptimo aproximado en viajes con vehículos sin trasferencia y el viaje privado. (Cao et al., 2021)

Finalmente observamos la utilidad de nuestros método de selección de documentos y la necesidad de estas 11 fuentes principales para el desarrollo de nuestra investigación.



# Marco teórico

## 3.1. Problema de enrutamiento de vehículos (VRP)

El problema de enrutamiento de vehículos, del inglés (*vehicle routing problem*, VRP), destaca por ser la generalización del problema del viajante (*travelling salesman problem*, TSP), un problema de optimización combinatoria muy recurrente en el mundo de los problemas NP-Hard, el cual busca la ruta más corta que un viajante puede recorrer una cantidad de ciudades y regresar al origen. («The Traveling Salesman Problem», 2008)

Este problema se planteó por primera vez en la literatura por Dantzig y Ramser en 1959, y buscaba encontrar una ruta óptima donde una flota de automóviles recorren estaciones de servicio fijas hasta volver al punto inicial o depósito. (Montoya-Torres et al., 2015)

Para poder resolver esta clase de problemas, se emplean distintas heurísticas y metaheurísticas para llegar a soluciones en un tiempo accesible.

### 3.1.1. Variantes del VRP

El problema VRP tiene algunas variables, que se acercan a problemas más cotidianos, algunas de las relevantes son las siguientes:

- **Problema de ruta de vehículo con visitas programadas (VRPPD):** el vehículo tiene que ir hasta un punto de recolección para llevar la mercancía a un punto de entrega basada en la capacidad del vehículo y tiempo de viaje.
- **Problema de enrutamiento del vehículo con ventanas de tiempo (VRPTW):** Las ubicaciones de entrega tienen ventanas de tiempo donde se deben realizar un número programado de entregas.
- **Problema de ruta de vehículo capacitado (CVRP):** El problema tiene cierta relación con el problema de la mochila y plantea una capacidad limitada de paquetes de transporte por envío, similar al VRPPD sin la necesidad de límite de tiempo.

- **Problema de enrutamiento de vehículos compartidos (VRPSD):** El problema busca encontrar una óptima ruta para un conjunto de vehículos compartidos para visitar un conjunto de puntos de entrega o recolección de mercancías, sujeto a restricciones de capacidad y tiempo de viaje. El objetivo es asignar pasajeros a vehículos de manera que se minimice el costo total de los viajes.

### 3.2. Viajes compartidos

Seattle Smart Traveler define al viaje como la acción de trasladarse de un punto geográfico a otro (Dailey et al., 1999). El viaje compartido, del inglés *ride-sharing*, es un viaje conjunto de al menos 2 participantes que comparten un mismo vehículo. Un viaje compartido exitoso requiere coordinación con respecto al tiempo y las especificaciones de recojo y despacho de un pasajero. Estas pueden necesitar de otros factores como el costo de viaje, reputación del conductor, tipo de vehículo, entre otros.

Un *viaje compartido no organizado* suele involucrar factores más cotidianos que involucran familiares y amigos, donde no se tiene claro el punto de destino de cada pasajero sino únicamente durante la ruta. Este tipo de viajes no son posibles de escalar, debido a la deficiencia de información inicial, que implica fallas en la estimación debido a la comunicación limitada.

Un *viaje compartido organizado* es operado bajo alguna agencia remota sin tener en cuenta alguna relación entre los pasajeros.

Debido a esto, el viaje compartido ofrece un potencial como servicio escalable. El acuerdo previo por parte de el proveedor del servicio y el pasajero es una característica clave en los viajes compartidos ofreciendo mayor confianza y credibilidad al servicio, a diferencia de un taxi común o transporte público de la calle. El acuerdo de viaje pueden comenzar cuando las solicitudes u ofertas de viajes se envían a través de Internet o por teléfono a los proveedores de servicios, que luego tienen como objetivo igualar la oferta y la demanda de viajes.

Existen 2 tipos de proveedores:

- **Servicio operador:** Operan bajo su propia flota de vehículos y conductores.
- **Agencia de emparejamiento:** Facilitan el servicio mediante emparejamiento con conductores independientes, usando el método de crowdsourcing.

Una característica notable es que la mayoría de las decisiones las toman los operadores de servicios, mientras que los participantes simplemente deciden si o no participar. Por el contrario,

las agencias de emparejamiento se centran en los servicios de emparejamiento de viajes entre conductores de automóviles y pasajeros individuales. A diferencia de los operadores de servicios, las agencias de emparejamiento no proporcionan vehículos ni conductores. En cambio, los conductores individuales tienen sus propios planes de viaje.

### 3.2.1. Tipos de viajes compartidos

El servicio de viajes compartidos tiene diferentes enfoques y de como evaluar, sin embargo, hablando únicamente de las empresas TNC podemos categorizarlas de la siguiente manera:

1. **Viaje compartido estático (Ride-Sharing Static):** En el ride-sharing estático, el número de pasajeros y la ruta se conocen previamente. Se suele considerar como un servicio donde los pasajeros pueden tener orígenes comunes y diferentes puntos de despacho. Por ejemplo, en Psaraftis (1980) se planteó un ride-sharing estático con el objetivo de minimizar el tiempo de duración del viaje, basándose en el problema del viajero (TSP).
2. **Viaje compartido dinámico (Ride-Sharing Dinamic):** El ride-sharing dinámico no tiene conocimiento de los pasajeros, su origen y destino, ni el número de vehículos disponibles. Toma los pedidos de forma que los vayan encontrando en su camino y con poca antelación.
3. **Coche compartido (carpooling):** A diferencia de las actividades de ride-sharing estático , el carpooling son actividades menos flexibles donde se realizan viajes fijos donde existe un único destino y varias personas con un solo punto de partida, su objetivo es transportar simultáneamente a varias personas desde un punto de partida común a un punto final común (Nechita et al., 2016).

En la tabla 3.1 podemos ver los pros y contras que demandan cada uno de estos tipos.

Tipo	Ventaja	Desventaja
Ride-Sharing Estático	Óptimo para conjuntos pequeños.	No escalable en tiempo real
Ride-Sharing Dinámico	No necesita conocer estado de los pasajeros y elige al mejor.	Es codicioso y carga con muchos casos
Carpooling	Ruta simple, libre de condiciones	Caso específico del ride-sharing estático y poco flexible

Cuadro 3.1: Tabla de ventajas y desventajas de los tipos de viajes compartidos.

### 3.2.2. Precios

Dado que la tarifa del viaje determina no solo el precio que el pasajero tiene que pagar por el viaje, sino también el ingresos para el conductor, las decisiones de fijación de precios influyen en las distribuciones de la demanda y la oferta a través del precio, por ejemplo, el uso de precios de aumento durante las horas pico.

Normalmente, los participantes en el sistema de transporte compartido están motivados por el hecho de compartir los costes del viaje, viajar rápido mediante el uso de vehículos grandes y mitigar los problemas medioambientales, en lugar de obtener beneficios. Además, dado que la forma final del transporte compartido sólo se determina cuando se determina el último pasajero de un vehículo, los participantes no pueden evaluar el valor del transporte compartido en el momento de presentar el pedido, a diferencia de las bolsas de valores y de anuncios. Por tal motivo, las agencias de emparejamiento, optan por ampliar la fijación de precios para evitar la falta de trazabilidad del cálculo de estos.

La fijación de precios especifica la cantidad de dinero que se transfiere entre las partes implicadas, incluyendo la forma de compartir los costes de la gasolina, el peaje y el aparcamiento, y la forma de cobrar las tasas de transacción por parte de las agencias de correspondencia. En la industria se utilizan los siguientes tres tipos de reglas de fijación de precios:

- **Precio de catálogo:** Los conductores o pasajeros especifican sus precios límite al hacer el listado. Por lo tanto, los propietarios del listado determinan su precio.
- **Precio basado en reglas:** El precio se determina mediante una fórmula de cálculo de costes especificada por la agencia. Normalmente, la fórmula es una tarifa estándar predeterminada por distancia multiplicada por una distancia calculada entre los lugares de recogida y entrega.
- **Precios basados en la negociación:** La agencia no participa en la fijación de precios. Se negocia entre los socios potenciales mientras determinan los lugares de recogida y entrega.

### 3.2.3. Patrones y variables

En la literatura, (Morency, 2007) clasificó al carpooling mediante elementos posicionales. Entre la presentación de estos elementos tenemos los más importantes:

- **Real distance  $dR_b$ :** Nos referimos a la distancia original del viaje tomado si hubiera un solo pasajero, para el pasajero  $b$ .

- **Relative distance**  $d_b$ : Nos referimos a la distancia relativa del viaje tomado por el pasajero  $b$  no necesariamente solo.
- **Rideshare route**  $R(a, B)$ : Ruta de viaje compartido. Donde  $a$  es el conductor de la ruta y  $B$  es el conjunto de pasajeros del viaje.
- **Pick-up location**  $pk_b$ : Punto de recojo de un pasajero  $b$ .
- **Drop-off location**  $dp_b$ : Punto de despacho de un pasajero  $b$ .

A continuación, presentamos algunos patrones de viaje más comunes en los viajes compartidos.

- Patrón 1 (viaje idéntico): En este es el flujo original del problema, donde el punto de destino del conductor y el punto de despacho del pasajero son los mismos, esto puede ser aplicado a más de un pasajero.

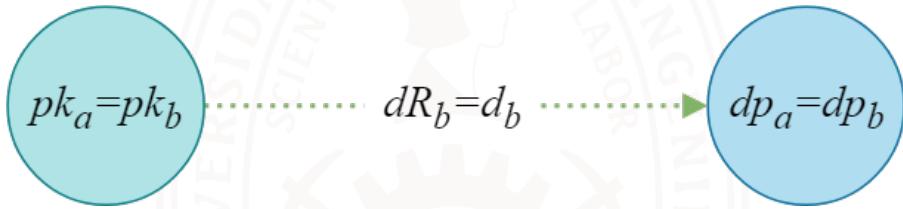


Figura 3.1: Ilustración del patrón 1, viaje idéntico.

- Patrón 2 (viaje incluido): Para este, el origen y destino de un pasajero  $b$  está en el camino del conductor  $a$ , quien tiene otra ruta con otro pasajero  $c$ , tenemos en cuenta además que la ruta del viaje de cada pasajero es distinta (es decir  $R(a, c) \neq R(a, b)$ ); para este caso la distancia real del viaje cambia para ambas rutas ( $dR_b, dR_c$ ), puesto que para el pasajero  $b$  sus puntos de recojo y destino están incluidos en la ruta del conductor  $a$  ( $pk_b, dp_b \in R(a, c)$ ), por lo que podemos decir que el recorrido real del pasajero  $c$  será la distancia recorrida por el conductor ( $dR_c = d_a$ ).

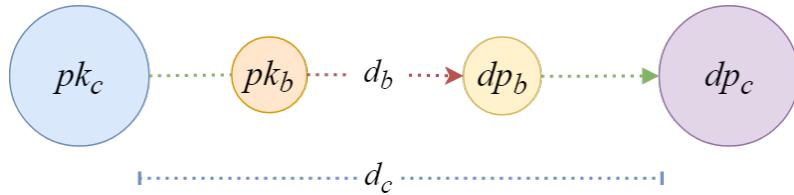


Figura 3.2: Ilustración del patrón 2, viaje incluido.

- Patrón 3 (viaje parcial): Dada la ruta de un conductor  $a$  con el pasajero  $c$ ,  $R(a, c)$ . Se presenta el pasajero  $b$  para el cual el punto recojo  $pk_b$  o el de entrega  $dp_b$  no están en la ruta, es

decir,  $pk_b \in R(a, c) \wedge dp_b \in R(a, c)$ ; por lo tanto existe dos puntos  $u_b, v_b \in R(a, c)$  y  $(u_b = pk_b) \vee (v_b = dp_b)$ , esto implica que no podemos asegurar que la distancia del pasajero  $c$  sea igual a la distancia recorrida por el conductor, debido a la existencia del nuevo pasajero  $b$ . El viaje es parcial porque el pasajero sigue la ruta una parte del viaje.

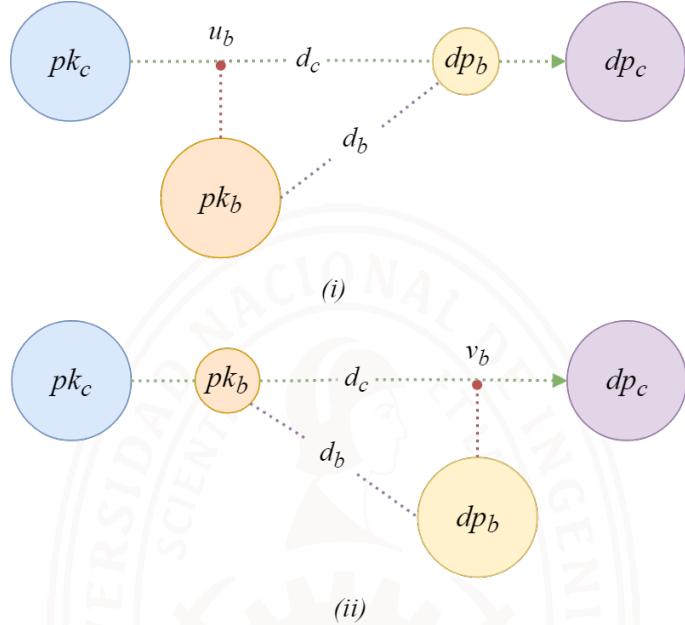


Figura 3.3: Ilustración del patrón 3, donde, (i).  $u_b = pk_b$ , (ii).  $v_b = dp_b$ .

- Patrón 4 (viaje por desvío): Finalmente el último y más común patrón de viaje, este sucede al momento de tomar al pasajero  $b$ , siendo que ninguno de sus puntos de recojo y entrega se encuentran en la ruta del conductor ( $pk_b, dp_b \in R(a, c)$ ), siendo que este tendrá que realizar un desvío en su ruta original para recoger y dejar al pasajero  $b$ . Esto generará distancias directas entre los puntos de inicio y partida de los pasajeros, y la distancia relativa será mucho más extensa.

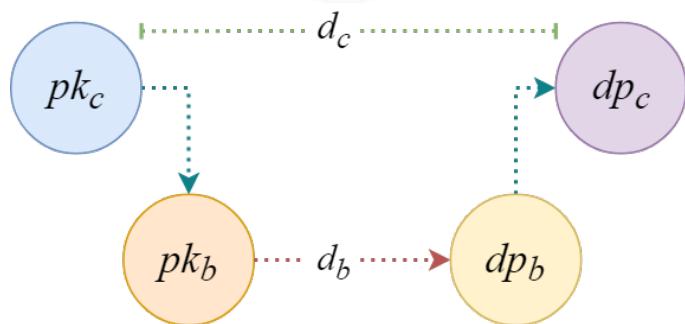


Figura 3.4: Ilustración del patrón 4, viaje por desvío.

### 3.2.4. Complejidad

Este problema tiene unas formas de solución pero podemos decir que el problema de viajes es un problema NP-hard basado en el problema del viajante, donde más adelante sabremos que es un camino Hamiltoniano.

#### 3.2.4.1. Prueba

Generalmente podemos transformar un problema de optimización dado en un problema de decisión relacionado. Para probar que sea un NP-hardness de un problema de optimización dado, podemos probar que el problema de decisión relacionado de este problema de optimización es NP-Completo. (Cormen et al., 2009) Es evidente que el problema de cálculo de la ruta ridesharing es un problema de optimización. Por lo tanto, si el problema de decisión relacionado del problema de cálculo es un problema NP-completo, entonces el problema es NP-Hard. El problema de decisión relacionado es, en grafo de puntos  $G = (B, E)$ , una solicitud de entrega  $R = (o, D, c, n)$ , poder determinar si existe un camino  $C' = (B', E')$  tal que la distancia media de llegada de todas las personas no sea superior al camino secuencial de los pedidos. A continuación, tenemos que demostrar es NP-hard. Es suficiente decir que, si un caso especial del problema de decisión es NP-hard, entonces el problema es NP-hard.

Consideramos un caso especial del problema de la decisión, dos vértices de la red de pedidos cualesquiera  $p_o$  y  $p_f$ , la longitud de un punto a otro es la misma, en otras palabras, el grafo viene de un grafo dirigido a una ruta no dirigida.

Si se demuestra que un problema es la reducción de un problema NP-hard conocido, entonces este problema es NP-hard. Demostraremos que problema es un problema de la camino Hamiltoniano, un problema NP-hard bien conocido.

Sea  $H = (B, V)$  ser un posible camino Hamiltoniano. Formamos un conjunto de aristas según un conjunto de decisión especial del problema  $E = e_{ij} | i, j \in B \wedge i \neq j$ , y definimos la la longitud de la arista según la distancia  $r_{ij}$  por:

$$r_{ij} = \begin{cases} 0 & e_{ij} \in V \\ 1 & e_{ij} \notin V \end{cases} \quad (3.1)$$

Se puede interpretar de cualquier arista en  $h$ , una parte de  $V$ , por lo que tiene el valor 0 en  $G$ . Además, siempre que el problema de la decisión especial tenga un plan de conducción  $h'$ , la distancia media de llegada es 0. Porque la longitud de la camino de cada arista en  $E$  es 0 o 1, cada arista tiene exactamente la longitud del camino 0. Por lo tanto, los aristas en  $h'$  sólo vienen de  $V$ . En consecuencia  $h'$  es un camino Hamiltoniano que comienza desde  $i$  en gráfico  $H$ . De este

modo, puede observarse que el problema de la decisión especial sobre el uso compartido es la reducción del problema de camino Hamiltoniano. Entonces el problema de cálculo de la ruta es NP-hard. ■

Dado que el problema de cálculo de la ruta de uso compartido del automóvil es NP-hard, el tiempo requerido para resolver el problema crece rápidamente junto con el aumento del número de pedidos sobre la misma ruta y el numero de vehículos presentes. Por lo tanto, es imposible encontrar un algoritmo óptimo para resolver este problema.

### 3.2.5. Beneficios y riesgos

Según Shaheen et al. (2018), el carpooling ofrece beneficios a largo plazo como la reducción del consumo del combustible, comprobando el aporte en reducir gases de efecto invernadero por los automóviles. Se comprobó que los vehículos que disponen viajes compartidos reducen el desplazamiento de autos en un 4 % disminuyendo el tráfico vehicular, además estos servicios abren las puertas a viajes más económicos y rápidos.

Entre sus deficientes esta la falta de privacidad y problemas en la salud pública por la alta probabilidad de contraer enfermedades virales; la revista Forbes este año realizó un análisis de la baja en la rentabilidad de los viajes compartidos por su falta de seguridad en la salud (Cristóbal, 2021), en este se especifica la pérdida de pasajeros en las agencias más populares (Uber y Lyft) de un 70 %, siendo la salud uno de su principal defecto en la actualidad.

## 3.3. Camino Hamiltoniano

En teoría de grafos, el camino Hamiltoniano, es una sucesión de aristas adyacentes que recorre todos los vértices del grafo una sola vez. Además el problema del ciclo Hamiltoniano es considerado un problema de NP-Completo de Karp. Donde también se lo define como un NP-hard. (Karp, 1972)

### 3.3.1. Definición formal

Sea  $G(V, E)$  un grafo con  $|V| = n$ , sin vértices aislados de aristas. Se denota un camino de Hamilton en  $G$  a todo camino que contiene los  $n$  vértices sin repetición.

### 3.3.2. Digrafo Hamiltoniano

El concepto de un camino Hamiltoniano dentro de un digrafo es muy similar, donde podemos mencionar:

- Es un camino dirigido Hamiltoniano si visita todos los vértices del digrafo una sola vez.
- Un digrafo que solo posee un camino Hamiltoniano se llama digrafo semi-Hamiltoniano.

### 3.3.3. Teorema de Ore y Dirac

Partiendo que todo grafo completo mayor a 2 vértices es Hamiltoniano. Se nos permite tener ciertas condiciones directas he indirectas de un grafo Hamiltoniano; las cuales se pueden apreciar en el Teorema de Ore y el Teorema de Dirac.

#### 3.3.3.1. Teorema 1 (Ore)

Si un grafo conexo  $G$  simple, con  $n \geq 3$ , en el cual se cumple que  $\text{grado}(u) + \text{grado}(v) \geq n$  para todo par de vértices no adyacentes  $u, v$  entonces  $G$  es Hamiltoniano. (Ore, 1960).

#### 3.3.3.2. Teorema 2 (Dirac)

Si un grafo de  $n$  vértices,  $n \geq 3$  donde  $\text{grado}(v) \geq n/2$  para todo  $v$  entonces  $G$  es Hamiltoniano. (Dirac, 1952)

De estos, podemos deducir algunos colorarios, donde se cumple la existencia de un camino Hamiltoniano infiriendo del teorema 3.3.3.2.

**3.3.3.2.1. Colorario** Si  $G$  es un grafo de  $n$  vértices,  $n \geq 3$ , en el cual  $\text{grado}(u) + \text{grado}(v) \geq n - 1$ , para todo par de vértices no adyacentes  $u, v$ , entonces  $G$  posee un camino Hamiltoniano.

### 3.4. Programación Entera y Lineal

La programación entera (IP) es el área de la optimización matemática en la que el problema a resolver está compuesto por variables y restricciones enteras; además, tienen una función objetivo, que les permite realizar problemas de maximización y minimización. La IP tiene diferentes aplicaciones y sobre todo destaca en optimización combinatoria y grafos. Al igual que un problema de programación lineal (LP), este se basa en un conjunto de restricciones que acotan el conjunto solución; sin embargo, a diferente un problema LP, estos tienen solución discreta. (Elmaghraby, 2003; Sutton y Barto, 2018)

Algunas de sus aplicaciones nos permiten resolver problemas como:

- Problema de la mochila (knapsack problem)
- Problema del viajante (travelling salesman problem)
- Problemas de asignacion (assignment problem)
- Problema de covertura (covering problem)

Como se puede ver, la optimización matemática que aplica la programación entera cuenta con numerosos ejemplos de problemas NP de teoría de grafos y combinatoria. (Wolsey, 2021)

### 3.5. Algoritmos Genéticos

Un algoritmo genético es un proceso de solución usando conceptos evolutivos neo Darwinistas. Esta enfocado en resolver problemas de optimización. En efecto, busca simular la evolución biológica representada por soluciones específica que cambian y adaptan según el método de biológicos inspirados por la selección natural. A continuación mencionaremos algunos agentes claves según Sivanandam y Deepa (2008) durante el proceso evolutivo:

- **Individuo:** Posible solución al problema.
- **Población:** Conjunto de soluciones o individuos.
- **Fitness:** Función de evaluación que indica la calidad del Individuo.
- **Genotipo:** Representación genética de los individuos compuesto por genes o características.

- **Fenotipo:** Representación numérica de los individuos generada por los genotipos.

Además, el concepto de algoritmo genético se basa en 4 métodos importantes para el desarrollo de mejores soluciones o individuos, estos son:

- **Mutación:** Método de alteración de características de un mismo individuo.
- **Cruzamiento:** Método de combinación de características de padres para generar hijos.
- **Selección de padres:** Método de selección de mejores padres para la generación.
- **Reemplazo de generación:** Método de selección de la nueva población.

Ahora pues todos estas variables nos dan la información necesaria para formar un algoritmo iterativo que nos permita aplicar el concepto evolutivo por cada grupo de individuos, el cual llamamos, *algoritmo canónico*. Como se puede ver en el algoritmo 1, un algoritmo canónico usa 2 criterios importantes:

- Recorrer un numero máximo de iteraciones que indica un máximo de generaciones posibles.
- Evaluar el problema hasta llegar a un óptimo local aceptable.

Mientras estos 2 puntos no se cumplan el ciclo de aprendizaje generacional se ira realizando iterativamente.

---

**Algorithm 1** Algoritmo Genético Canónico

---

```

 $t \leftarrow 0$ 
 $P_t \leftarrow Poblation(size)$ 
 $eval(P_t)$                                 ▷ Evaluación de fitness de todos los individuos de  $P_t$ 
while  $t < Generation$  do
     $D_t \leftarrow selection(P_t)$                 ▷ Selección de padres respecto a  $P_t$ 
     $crossover(D_t, p_{crossover})$                   ▷ Evaluación de crusa de padres  $D_t$ 
     $mutation(P_t, p_{mutation})$                     ▷ Evaluación de mutación del conjunto  $P_t$ 
     $generate(P_t)$                                 ▷ Selección de nueva población
     $eval(P_t)$ 
     $t \leftarrow t + 1$ 
end while
return  $best(P_t)$ 

```

---

### 3.6. Aprendizaje por refuerzo

Aprendizaje por refuerzo, en inglés (reinforcement learning, RL) pertenece a una de las categorías de la Inteligencia Artificial. El enfoque de RL es capacitar a un agente de manera que este pueda optimizar su comportamiento acumulado y aprendiendo de sus experiencias mientras interactúa con un entorno. El óptimo del modelo depende de la maximización de la recompensa total acumulada por el agente. (Sutton y Barto, 2018)

RL se basa en el marco del proceso de decisión de Markov (MDP), donde nuestro agente tiene un estado  $s$  de un conjunto de estados  $S$  y puede realizar acciones  $A$  definida para interactuar con el entorno.

Durante el proceso de ejecución, se lanzan acciones al entorno constantemente. Una vez ejecutada la acción, el agente recibirá una recompensa inmediata  $R(s, a)$  del entorno y su estado cambia de acuerdo a los parámetros de transición, este proceso se repetirá hasta obtener un estado terminal  $s_f$ , según la figura 3.5 dado el control de los tripletes  $(s_t, a_t, r_t)_t$ . Donde  $t$  es el índice de la iteración. El objetivo del MDP es maximizar la recompensa acumulada, por lo que, se toma de variable a

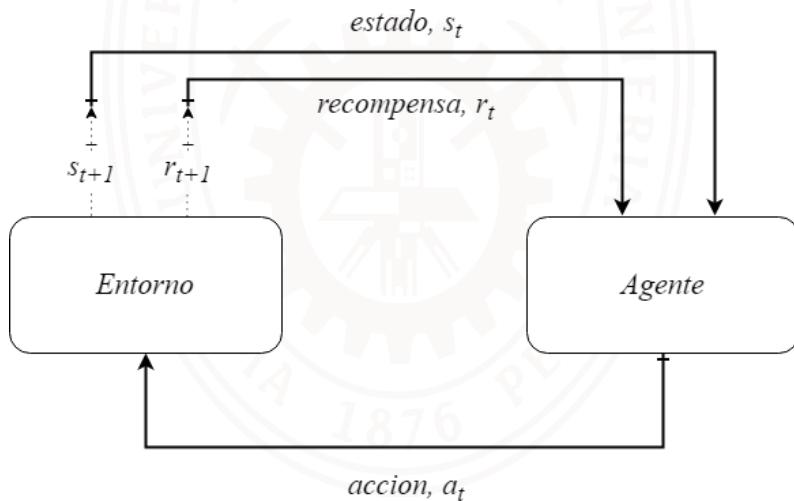


Figura 3.5: Ilustración de la interacción entorno-agente.

la función de valor que representa la esperanza de la recompensa acumulada respecto al estado  $s$  hasta el final del episodio.

$$V(s) = \mathbb{E}[\sum_k^n \gamma_k r_t | S_t = s_k] \quad (3.2)$$

Donde en la ecuación 3.2,  $\gamma$  es el factor de descuento ( $\gamma \in [0, 1]$ ). Además, esta función de valor satisface la ecuación de Bellman la cual busca la recompensa acumulada hasta el estado  $s_t$

$$V(s_t) = \sum_{a_t} P(s_{t+1}, r_t | s_t, a_t) (r_t(s_t, a_t) + \gamma V(s_{t+1}))$$

$$V(s) = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})|S_t = s] \quad (3.3)$$

Teniendo debemos reconocer la importancia de que nuestro agente debe aprender a priorizar la mejor acción posible según los estados posibles.

Con esto damos paso a los algoritmos Q-Learning, los cuales intentan aprender a largo plazo la recompensa futura según los estados y acciones. Para la función Q valor, la función actúa bajo la acción y valor

$$Q(s, a)^\pi = \mathbb{E}\left[\sum_k^n \gamma_k r_t | S_t = s, A_t = a\right] \quad (3.4)$$

donde la política es una función  $\pi(s) : \mathcal{S} \rightarrow \mathcal{A}$ .

Teniendo todo esto la iteración de valores se utiliza la ecuación óptima de Bellman y ayuda a encontrar el siguiente estado. Es decir, calculamos la nueva función de valor de un estado conectando con el valor de la iteración anterior en la ecuación óptima de Bellman.

$$V(s) \leftarrow \max_a \sum_{s',r} P(s',r|s,a)(r + \gamma V(s')) \quad (3.5)$$

Podemos representar a la ecuación 3.2 usando la forma de la ecuación de Bellman (3.6) en su forma iterativa. A partir de esto podemos armar un método sin modelos que aprenda bajo una función de valor y se optimiza la política sin alterar el entorno. El ejemplo más común es el temporal-difference learning (TD) que realiza actualizaciones iterativas de  $V$ .

$$V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s)) \quad (3.6)$$

donde  $\alpha$  es la tasa de aprendizaje ( $\alpha \in [0, 1]$ ), que indica la rapidez de convergencia.

### 3.6.1. Q-Learning

Ahora bien, si este valor de aprendizaje dependiera de las acciones posibles por un estado, dando mayor control al objetivo de la recompensa, da como resultado a Q-learning, que se basa en actualizar la función Q-valor .

$$Q'(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_a Q(s', a) - Q(s, a)) \quad (3.7)$$

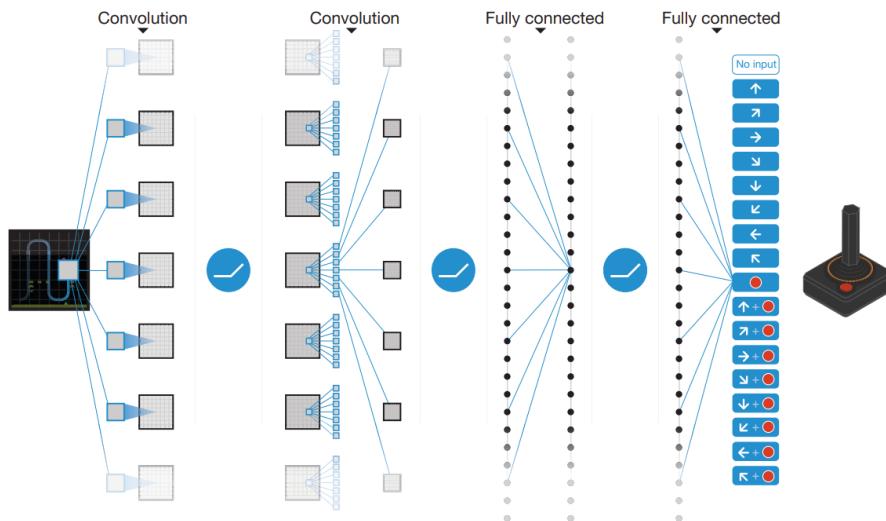
En la búsqueda de poder realizar operaciones más complejas usando el formato de entrenamiento de las redes neuronales, se planeó el aprendizaje por refuerzo profundo usando el concepto iterativo de los Q-valor el cual se propuso para la solución de distintos juegos de Atari reconocidos, naciendo el deep Q-network, DQN. (Mnih et al., 2015) El DQN aproxima un  $Q(s, a)$  por una red neuronal a lo largo de algunas heurísticas que ofrecen estabilidad al entrenamiento. Un método basado en políticas aprende directamente calculando el gradiente de política (PG), el

gradiente de la recompensa acumulada con respecto a los parámetros de política  $\theta$ .

$$\nabla_{\theta} \mathcal{J}(\theta) = \sum_s \mu(s) \sum_a Q^{\pi}(s, a) \nabla_{\theta} \pi(a|s, \theta) \quad (3.8)$$

donde  $\mu$  es la distribución de la política en  $\pi$ .

Finalmente, podemos ver en la figura 3.6 una arquitectura clásica de una DQN basada en el aprendizaje del juego de Atari que contiene 2 capas de convolución para el reconocimiento de características y 2 capas totalmente conectadas, que muestran finalmente una posible acción aplicando una función exponencial normalizada o función softmax.



**Figura 3.6:** Ilustración esquemática de la red profunda usada para el aprendizaje por refuerzo profundo del videojuego de ATARI. (Mnih et al., 2015)

# Metodología

En este capítulo, se explican las metodologías adoptadas para generar el análisis de datos y la arquitectura del modelo.

## 4.1. Base de datos

Para poder realizar nuestro modelo debemos tener datos que puedan explicar perfectamente la información necesaria en el espacio. Para ellos buscamos utilizar datos que ofrezcan información en el formato estándar de representación geográfica, el formato GeoJson y el estándar de GIS, que ofrece información geográfica y cartográfica (Hugh-jones, 1997).

### 4.1.1. Conjunto de datos Chicago Taxi Trips

#### 4.1.1.1. Resumen

El conjunto de datos de estudio (Chicago Portal, 2021) es un reporte de viajes en taxi en la ciudad de Chicago, bajo un proceso de regularización y conteo de taxis en la ciudad. Todas las empresas afiliadas a estos servicios registran sus taxis mediante un identificador en la data pública de viajes de taxis de Chicago para proteger la privacidad.

El conjunto de datos de Chicago Taxi Trips fue lanzado desde enero del 2013 hasta la fecha, sin embargo según el reporte de actualizaciones del conjunto de datos de Chicago, en octubre del 2017 desactivó la subida de nuevos datos por problemas en la recolección de datos, hasta abril del 2019 que reanudó el reporte de datos (Chicago Data Portal, 2019).

#### 4.1.1.2. Justificación

Sé tomaron los datos de Chicago Taxi por su uso de datos espaciales SQL, los cuales representan información sobre la ubicación física y la forma de los objetos geométricos. Estos objetos pueden

ser ubicaciones puntuales u objetos más complejos como carreteras, etc. Esto nos permitirá de forma sencilla obtener la información de viajes privados y generar viajes compartidos viables en un contexto real de viajes. (Microsoft, 2020)

#### 4.1.1.3. Estructura del conjunto de datos

El conjunto de datos posee 23 columnas los que se describen brevemente en cuadro 4.1.

Columna	Descripción	Tipo de dato
Trip ID	Identificador único para el viaje.	Texto
Taxi ID	Identificador único para el taxi.	Texto
Trip Start Timestamp	Fecha de inicio del viaje.	Timestamp
Trip End Timestamp	Fecha final del viaje.	Timestamp
Trip Seconds	Tiempo del viaje en segundos.	Número
Trip Miles	Distancia del viaje en millas.	Número
Pickup Census Tract	Sección censal donde comenzó el viaje.	Número
Dropoff Census Tract	Sección censal donde terminó el viaje.	Número
Pickup Community Area	Área Comunitaria donde comenzó el viaje	Número
Dropoff Community Area	Área Comunitaria donde terminó el viaje.	Número
Fare	Tarifa del viaje.	Número
Tips	Propina para el viaje.	Número
Tolls	Peajes para el viaje.	Número
Extras	Cargos adicionales por el viaje.	Número
Trip Total	Cargos adicionales por el viaje.	Número
Payment Type	Tipo de pago del viaje.	Texto
Company	Compañía de taxis.	Texto
Pickup Centroid Latitude	Latitud del punto de recogida.	Número
Pickup Centroid Longitude	Longitud del punto de recogida.	Número
Pickup Centroid Location	Ubicación espacial del punto de recogida.	POINT
Dropoff Centroid Latitude	Latitud del punto de abandono.	Número
Dropoff Centroid Longitude	Longitud del punto de abandono.	Número
Dropoff Centroid Location	Ubicación espacial del punto de entrega.	POINT

**Cuadro 4.1:** Tabla de estructura del conjunto de datos Taxi Trip.

#### 4.1.1.4. Limitaciones

Aun teniendo una estructura completa, algunos pedidos de taxis tienen información incompleta para ciertas ocasiones, entre todas mencionamos algunas de ellas:

- Cuando el punto de recojo o destino se encuentran fuera de la ciudad de Chicago, toda información relacionada al punto se encuentra vacía.
- Los datos de los precios en algunos casos se encuentran fuera de un precio razonable superando los 300\$ por un viaje de 6 millas.
- El **Pickup Census Tract** y **Dropoff Census Track** se encuentras descontinuados y no se usan desde 2019.

Además de ello, el conjunto de datos tiene actualizaciones mensualmente, y no en tiempo real.

### 4.1.2. Boundaries Community Areas

#### 4.1.2.1. Resumen

El conjunto de datos de estudio es una lista los límites actuales del área comunitaria de Chicago. Estos límites están representados por varias sub áreas, las cuales representan un sector del mapa de Chicago, el cual no ha cambiado en los últimos 4 años. (Chicago Data Portal, 2013).

#### 4.1.2.2. Justificación

Se plantea el uso de este conjunto de datos para reconocer en que área se encuentran los pedidos, ya que, utiliza un tipo agrupamiento de datos espaciales SQL llamado **MultiPolygon** haciendo más fácil el uso de la información espacial y realizar alguna reglas para el algoritmo.

#### 4.1.2.3. Estructura del conjunto de datos

El conjunto de datos tenía 10 columnas que se describen brevemente en cuadro 4.2.

Columna	Descripción	Tipo de dato
the_geom	Recuadro del área de la comunidad.	MULTIPOLYGON
PERIMETER	Perímetro del área de la comunidad.	Número
AREA	Área del área de la comunidad.	Número
COMAREA_	Sin Descripción.	Número
COMAREA_ID	Sin Descripción.	Número
AREA_NUMBE	Número asignado a cada área de comunidad.	Número
COMMUNITY	Nombre de cada área de comunidad.	Texto
AREA_NUM_1	Sin Descripción.	Número
SHAPE_AREA	Área perimetral del recuadro.	Número
SHAPE_LEN	Distancia perimetral del recuadro.	Número

Cuadro 4.2: Tabla de estructura del conjunto de datos Boundaries Community Areas.

#### 4.1.2.4. Observaciones

A pesar que el conjunto de datos este completo, existe información innecesaria y sin utilidad para la investigación como lo es **COMAREA\_**, **COMAREA\_ID** y **AREA\_NUM\_1**.

## 4.2. Preprocesamiento

Buscamos eliminar parte de las filas y columnas que no nos sirven, para ello descartamos las columnas que menos relevancia nos dará para toda la investigación, planteamos una limpieza de datos según las siguientes reglas a continuación:

- Los precios de los viajes no deben superar un precio lógico, usamos los 100\$ como el valor máximo del costo de un viaje.
- Dividimos las tablas según los identificadores de *Trip*, *Taxi* y *Point*.
- Un *Trip* no puede tener un punto de recojo o destino vacío.
- Al precio final del viaje debemos discrepar valores como extras o peajes ya que son arbitrarios y escapa de la estimación de precios.
- El tipo de pago no importa en el análisis.
- La sección censal (Census Tract) de los puntos de recojo y destino es información desactualizada ya que usamos las áreas comunitarias zonificar los puntos.

Finalmente dividimos ambos conjuntos de datos antes mencionadas (sección 4.1) en tablas organizadas entendibles para el uso de distintos modelos de optimización la cual representaremos con un modelo de base de datos relacional como se ve en la figura 4.1.

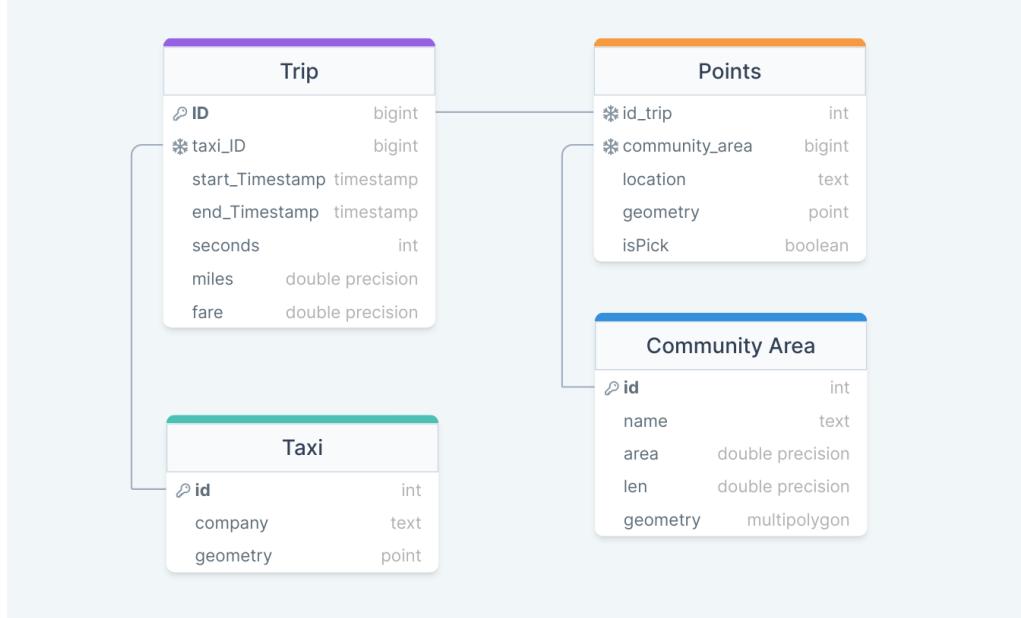


Figura 4.1: Grafica del modelo de base de datos.

Finalmente, realizamos un filtro de los datos más actuales de viajes de taxis post pandemia y lo limpiamos, para esto, optamos por realizar lo siguiente:

- En el conjunto de datos Taxi Trips eliminamos todos los datos que no tengan **Pickup Centroid Location** y **Dropoff Centroid Location**, esto debido a que son necesarios para generar un tabla con valor geométrico.
- Como podemos ver que cada tabla se destaca por un tipo de datos geometry, almacenada en una columna con su mismo nombre, esto ya que PostgreSQL junto a su extensión PostGIS, que nos permite tener un tipo de dato espacial haciendo más rápida la consulta, evitando así la carga de datos en memoria que resulten costosos de tratar, como el formato csv.
- Usamos llaves foráneas para verificar que ningún dato este sin usar.

### 4.3. Tecnologías

Como podemos ver en las secciones anteriores, requerimos de varias tecnologías y herramientas para llevar acabo este proyecto, las cuales se hace mención de las más importantes.

### 4.3.1. Docker

Docker es una plataforma de código abierto que permite implementar y administrar aplicaciones en "contenedores", los cuales son los que permiten virtualizar una aplicación libre de control de recursos de hardware. Se destaca su uso gracias a la portabilidad y ligereza de consumo como método de crear contenedores además de su fácil despliegue, evitando problemas de compatibilidad de recursos entre aplicaciones. (Docker, 2022).

#### 4.3.1.1. Docker Compose

La tecnología de contenedores se plantea por 3 categorías de implementaciones diferentes. Entre ellas se encuentra la Orquestación, el cual es la forma en que podemos administrar diferentes aplicaciones en varios contenedores, este se llama Docker Compose.

Entre sus características más destacadas tenemos su capacidad de persistencia de volúmenes o datos y bajo costo en recursos de computo. (IBM Cloud Education, 2022).

#### 4.3.1.2. Justificación

Se planteó usar desde un inicio la tecnología de contenedores por orquestación, ya que nos permite desplegar algunos servicios que necesitamos sin necesidad de perder información ni preocuparnos por la instalación previa, siempre que tengamos la imagen Docker que lo tenga instalado.

### 4.3.2. PostgreSQL

PostgreSQL o Postgres es un sistema gestor de base de datos relacional de código abierto muy popular y de fácil implementación, que usa una sintaxis extendida del lenguaje SQL junto con características destinadas a la manipulación de grandes cantidades de datos. (Group PostgreSQL Global Development, 2022).

#### 4.3.2.1. PostGIS

PostGIS es una extensión de PostgreSQL que agrega a Postgres la capacidad de procesar, transformar y analizar la información espacial de una base de datos dentro de esta, destacando su rapidez al operar objetos espaciales; además de importar y exportar información de formatos

relacionados a la información espacial (GeoPackage, ShapeFiles, SpatialLite, GeoJson, etc). (PostGIS, 2022).

#### **4.3.2.2. Justificación**

Usamos Postgres y PostGIS por su rapidez en este formato espacial, siendo el el mejor base de datos espacial según la especificación de *Simple Features for SQL* (SPSQL) siendo compatible con el formato de la **Open Geospatial Consortium** (OGC). Agregado a esto, Postgis es compatible con librerías de Python que hacen más fácil la operación. (Open Geospatial Consortium, 2022).

#### **4.3.3. OpenStreetMap**

OpenStreetMap (OSM) es una base de datos de mapas gratuitos de todo el mundo que alberga información geoespacial open source. Los recursos informáticos y de red son donados por universidades y organizaciones. Cabe aclarar que el conjunto de datos es información presentada por la comunidad, por lo que, **el mapa aún no está terminado** y presente continuas actualizaciones en tiempo real. (OpenStreetMap contributors, 2017).

##### **4.3.3.1. Justificación**

OSM nos presenta información de mapas de sectores del mundo, en formato geoespacial, útil para observar el mapa de Chicago y poder ofrecer resultados reales según el estado de las carreteras de la ciudad.

#### **4.3.4. Librerías de Python**

Python es catalogado como un lenguaje de programación más famosos y simples destacando su versatilidad y funcionalidad al momento de programar. Entre estas características, destacada su diversa cantidad de librerías, enfocadas a diferentes fines como aprendizaje automático, aprendizaje por refuerzo, Web,etc.

En nuestro caso usamos librerías enfocadas a la ilustración, geoespacial y aprendizaje por refuerzo. Siendo las más usadas las siguiente:

- **Networkx:** Paquete relacionado a la creación, manipulación de datos de redes complejas (grafos, digrafos, caminos,etc). (Hagberg et al., 2008).

- **GeoPandas:** Al igual que PostGIS, GeoPandas es un extensor de Pandas que agrega operaciones y estructuras espaciales y tipos geométricos. (Jordahl, 2014).
- **Folium:** Es un paquete que permite visualizar datos Geográficos espaciales en un mapa interactivo integrados a APIs de mapas, como OpenStreetMap y MapBox. (python-visualization, 2020).
- **SQLAlchemy:** ToolKit de SQL para Python conocido por tener un object-relational mapper (ORM), y permite operar consultas como cliente a una base de datos usando su enfoque *orientado a la complementariedad*. (Bayer, 2012).
- **OSMnx:** Paquete que permite descargar datos geoespaciales de OpenStreetMap, visualizar y analizar las redes de calles en todo el mundo según lo que se requiera. (Boeing, 2017).
- **PuLP:** Paquete de programación lineal enfocado en modelar problemas de optimización permitiendo operar basandonos en una función objetivo, variables y restricciones. (Iain Dunnin et al., 2011)
- **PyTorch:** Librería enfocada al cálculo tensorial comúnmente usado para el desarrollo de redes neuronales y el Deep Learning. (Paszke et al., 2019).

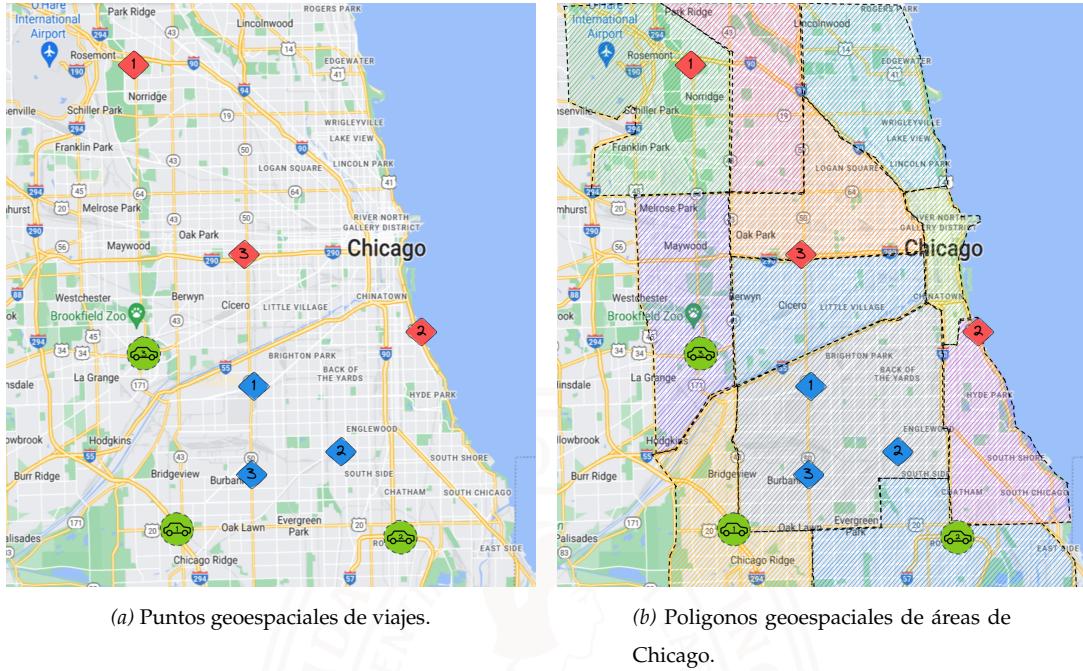
## 4.4. Planteamiento

Para poder hacer un planteamiento adecuado debemos describir la naturaleza y los casos que toma en cuenta el modelo esperado. A continuación presentamos el planteamiento ilustrativo y matemático del problema de viajes compartidos.

Partimos con la idea que un viaje compartido posee la capacidad de crear una sola ruta basandonos en varias rutas, y así unir varios pedidos en una sola ruta. Además, dada la naturaleza de la baja capacidad de los vehículos, autos privados, en este trabajo tomamos un máximo de 3 pasajeros por ruta. Sea en un momento de tiempo, 3 pasajeros solicitan un viaje a sus respectivos destinos, y existen 3 vehículos cercanos a estos puntos como vemos en figura 4.2a y tenemos conocimiento que cada punto pertenece a una área dentro de la ciudad (subsección 4.1.2).

Intersectando los datos geoespaciales como vemos en la Figura 4.2b, los puntos se encuentran en zonas diferentes de la ciudad.

En este escenario, se nos presenta 2 formas de resolver el problema. La primera, usando el transporte privados, podemos asignarle 1 vehículo a cada pasajero y dejarlos en su destino dejando



**Figura 4.2:** Ilustración de mapa junto con datos geográficos de viajes.

al pasajero en un tiempo promedio medio como se capta en la Figura 4.3a. En cuanto a la segunda solución, la Figura 4.3b nos ilustra una forma de generar una sola ruta según la cercanía de los pedidos respecto a los vehículos y sus destinos, siendo esta la forma en que se realiza un viaje compartido.

Evidentemente, en el caso de los viajes compartidos, según la cantidad de vehículos disponibles, existirá una lista de rutas posibles, donde finalmente solo uno de ellos será la mejor ruta para nuevos pasajeros, tomando al más apto, como se ve en la figura. 4.4.

Por eso, la forma práctica de representar esta red de rutas será mediante un grafo, donde nos encontramos con 2 tipos de redes según la tarea que buscamos resolver, como en la Figura 4.5.

Ahora, debemos considerar que toda conexión entre estos puntos posee una dirección, la cual dependerá de que puntos estén involucrados, donde tendremos ciertas reglas:

- La ubicación del conductor solo puede ser un punto de inicio sin retorno.
- Los puntos de recojo solo pueden conectarse a sus destinos como puntos de inicio sin retorno.
- Los puntos de recojo pueden conectarse a destinos de otros pedidos en ambas direcciones.
- Tanto los puntos de recojo y destino pueden conectarse en ambas direcciones entre si.

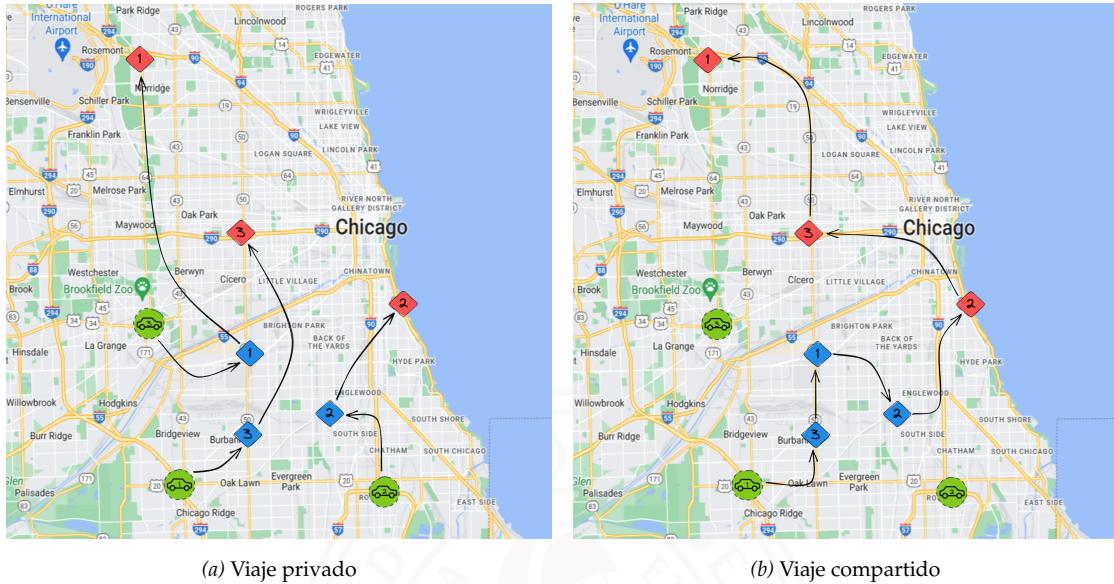


Figura 4.3: Ilustración de posibles rutas tomadas para el mismo problema de viajes

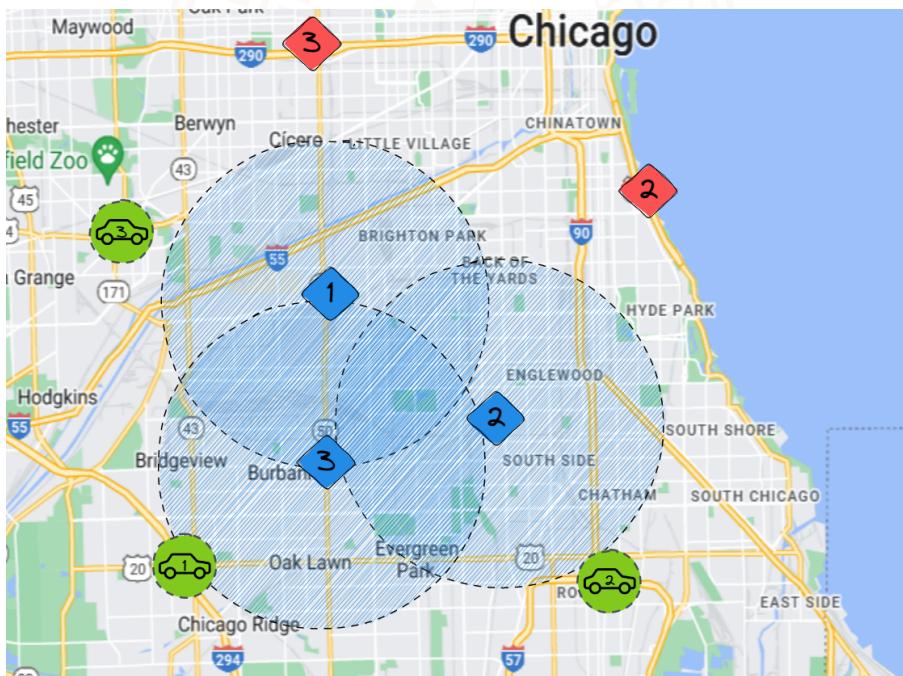


Figura 4.4: Ilustración de selección del mejor vehículo por cercanía a los puntos.

Con estas reglas podemos ver la forma visual de la red conectada de puntos de la figura 4.5a, y la red incluida el punto del vehículo en la figura 4.5b, donde cada conexión presente posee un peso o valor, que indica que tan difícil o fácil es moverse de un punto del mapa a otro.

Para esto, vemos 2 variables presentes: distancia geográfica y entre áreas comunitarias donde se encuentran los puntos, según sea de un vehículo a un punto de recojo (Fig. 4.6a), como de

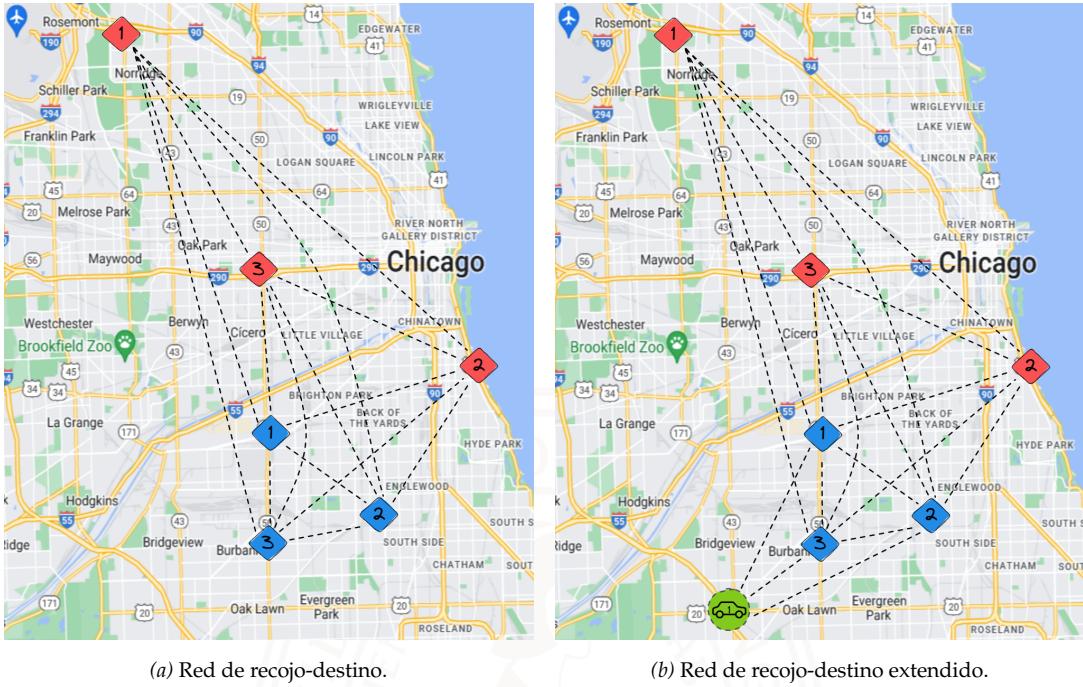


Figura 4.5: Ilustración de puntos geoespaciales conectados.

un punto de recojo a su destino (Fig. 4.6b). Esta relación nos muestra que, mientras más lejos estén los puntos (*distance*) y mayor cantidad de cruce entre áreas comunitarias (*cross\_community*) nuestro viaje por ese camino será más difícil.

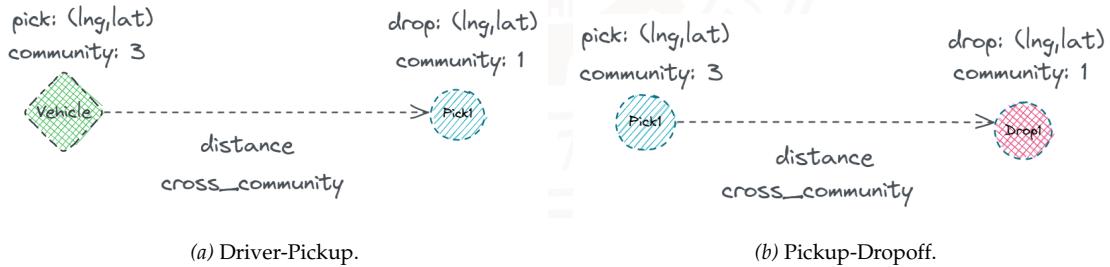


Figura 4.6: Ilustración de aristas según el tipo de vértices.

En resumen, un viaje compartido dispone de diferentes enfoques y casos, en este, usamos 3 pedidos que son recogidos por un solo vehículo para poder generar una red de recojo-destino que nos servirá para identificar cuál es el mejor orden para recoger y dejar nuestros pasajeros según las variables de distancia y área comunitaria. Con todo esto, tenemos lo necesario para tener el precio resultante. En el siguiente apartado definimos matemáticamente lo ya presentado para facilitar el planteamiento de las soluciones.

#### 4.4.1. Planteamiento matemático

Iniciamos mostrando el grafo generado por la figura 4.5:

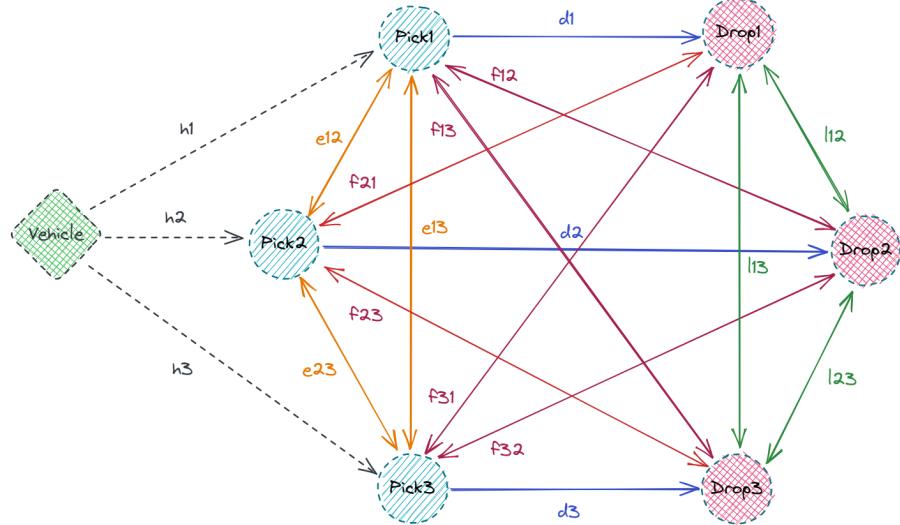


Figura 4.7: Ilustración de grafo de red de recojo-destino extendido.

Sea un digrafo ponderado  $G = (V, E, \omega)$  donde  $V$  es el conjunto de vértices donde cada  $v$  representa un par de coordenadas geográficas  $(\text{lng}, \text{lat})$  (Ec. 4.1).

$$V = \{v_i = (\text{lng}, \text{lat}) / v_i \in \mathbb{R}^2\} \quad (4.1)$$

Para el caso de del conjunto de aristas  $E$  es un conjunto no completo, es decir  $|E| \leq |V|^2$ , ya que su comportamiento varía según el tipo de punto donde te diriges.

$$E(v_i, v_f) = \{e = \overrightarrow{v_i v_f} / v_i, v_f \in V\} \quad (4.2)$$

Asimismo, definimos los actores del problema como un conjunto  $R$  de  $m$  pedidos ( $r_i \in R$ ) con puntos de recojo  $pick_i$  y despacho  $drop_i$  (Ec. 4.3), conociendo que,  $|R| \leq 3$ . Además, un conjunto  $H$  de  $n$  conductores ( $h \in H$ ) donde el conjunto tiene un punto de ubicación  $h_k$  (Ec. 4.4).

$$R = \{r_i = (pick_i, drop_i) : pick_i, drop_i \in V\} \quad (4.3)$$

$$H = \{(h_k) : h_k \in V\} \quad (4.4)$$

Seguidamente, definimos el conjunto de áreas comunitarias geográficas, *boundaires community areas* (BCA, 4.1.2), donde cada área  $\varepsilon$  es un polígono :

$$BCA = \{\varepsilon = [p_0, \dots, p_n] / p_i, \overline{p_i p_{i+1}} \in \mathbb{R}^2, i \in [0, n]\} \quad (4.5)$$

Por otro lado, usamos una función calificadora  $w : E \rightarrow \mathbb{R}^+$ , que evalúa cada arista. De este modo, para poder obtener el peso de una arista, se usa la recta que divide ambos puntos para determinar distancia y la cantidad de áreas que cruza. Con esto vemos conveniente, usar una relación multiplicativa, ya que a mayor cantidad de áreas el punto se encuentra más lejos, empero, de encontrarse en una misma área el valor multiplicativo será neutro, esto para darle más importancia a los puntos más cercanos entre ellos. Por lo que definimos una función  $\psi : E \rightarrow \mathbb{N}$ :

$$\psi(E(v_1, v_2)) = |E(v_1, v_2) \in BCA| \quad (4.6)$$

Ahora, definimos el conjunto que aristas que pueden existir debido que en el problema de viajes compartidos, no existe retorno a punto de recojo ni el retorno al punto de inicio del vehículo, entonces usamos los conjuntos  $R$  (Ec. 4.3) y  $D$  (Ec. 4.4), para definir el siguiente conjunto  $\Omega$ :

$$\Omega = \{ E(v_i, v_f) : v_f \notin d_j \vee i \neq f \vee (v_i, v_f \in r_k \wedge v_i \neq r_{k,drop}) \} \quad (4.7)$$

Usando la ecuación 4.6, podemos ver que  $\psi(E(v_1, v_2)) \geq 1$ , debido a que siempre existirá una área que contenga ambos puntos, con esto obtenemos la función  $w$ :

$$w(v_1, v_2) = \begin{cases} v_1, v_2 \in \Omega & \sqrt{(v_1.lng - v_2.lng)^2 + (v_1.lat - v_2.lat)^2} \cdot (\psi(E(v_1, v_2))) \\ v_1, v_2 \notin \Omega & 0 \end{cases} \quad (4.8)$$

En consecuencia, viendo el colorario 3.3.3.2.1, tomamos el grafo  $D$  que representa lo ilustrado en la figura 4.7 para demostrar que posee un camino Hamiltoniano.

**Demostración:** Sea el grafo  $D$  con  $|V| = 7$ , usamos 2 puntos no adyacentes,  $v_{vehicle}$  y  $v_{drop,i}$  y los grados representan la cantidad de conexiones,  $grado(v_{vehicle}) = 3$  y  $grado(v_{drop,i}) = 5$ . Finalmente,  $grado(u) + grado(v) \geq n - 1 \implies 3 + 5 \geq 7 - 1 \implies 8 \geq 6$ . Donde el grafo  $D$  tiene una camino Hamiltoniano ■

Finalmente, le debemos expresar a nuestro grafo de una forma entendible para los diferentes algoritmos que los puedan usar, para esto definiremos una matriz especial que defina su matriz de adyacencia, el cual tendrá el siguiente patrón:

$$Matrix_{ext}(G) = \begin{pmatrix} 0 & h_1 & 0 & h_2 & 0 & h_3 & 0 \\ h_1 & 0 & d_1 & e_{12} & f_{12} & e_{13} & f_{13} \\ 0 & d_1 & 0 & f_{21} & l_{12} & f_{31} & l_{13} \\ h_2 & e_{12} & f_{21} & 0 & d_2 & e_{23} & f_{23} \\ 0 & f_{12} & l_{12} & d_2 & 0 & f_{32} & l_{23} \\ h_3 & e_{13} & f_{31} & e_{23} & f_{32} & 0 & d_3 \\ 0 & f_{13} & l_{13} & f_{23} & l_{23} & d_3 & 0 \end{pmatrix} \quad (4.9)$$

Donde:

- $h_i$ : Representa las distancias del vehículo a cada punto de recojo  $i$ .
- $d_i$ : Representa las distancias de los puntos de recojo a sus destinos originales.
- $e_{ij}$ : Representa las distancias entre el punto de recojo  $i$  al punto de recojo  $j$ , donde  $i \neq j$ .
- $f_{ij}$ : Representa las distancias entre el punto de recojo  $i$  al punto de destino  $j$ , donde  $i \neq j$ .
- $l_{ij}$ : Representa las distancias entre el punto de destino  $i$  al punto de destino  $j$ , donde  $i \neq j$

Para obtener la matriz de adyacencia de únicamente los pedidos, es suficiente con eliminar la primera fila y columna de la matriz de  $G_{ext}$ , es decir:

$$\text{Matrix}(G) = \text{Matrix}_{ext}(G)[1 : , 1 :] \quad (4.10)$$

## 4.5. Definición del modelo matemático

Para modelar el problema vamos a plantear una variante del problema VRP con sus respectivas restricciones, variables y función objetivo, con el fin de implementar una optimización matemática de programación entera.

### 4.5.1. Variables de decisión

- $X_{ij}$ : indica que el vehículo se mueve de un punto  $i$  a  $j$  (0 o 1).
- $V_i$ : indica si el vehículo ya visitó el punto  $i$  (0 o 1).

### 4.5.2. Función objetivo

Buscamos minimizar el costo de los viajes, asumiendo que los pesos de la matriz están colocados correctamente, será representado por la suma de costos de los arcos incluidos en la ruta.

$$\text{Min}(Z) = \sum_i \sum_j c_{ij} \cdot x_{ij} \quad (4.11)$$

Donde  $c_{ij}$  es el peso de la arista  $(i, j)$ .

#### 4.5.3. Restricciones

Existen varias restricciones las cuales se derivan en 3 principales, capacidad, continuidad y recolección, estas pueden ser descritas dentro de las siguientes restricciones:

- Todos los puntos deben ser visitados durante el recorrido:

$$\sum_i^n v_i = 1$$

- Los puntos de recojo solo pueden ser visitados una vez:

$$v_i \leq \sum x_{ij} , \forall j$$

- Todo punto de entrega sólo puede ser visitado después de haber visitado su punto de recojo ( $i, j$ ):

$$v_j \leq v_i$$

- Solo se visitan los puntos de entrega que ya pasaron por su recojo, no necesariamente de forma consecutiva y cumplen con estas 3 condiciones:

$$\sum_i \sum_j x_{ij} \leq v_i$$

$$\sum_i \sum_j x_{ij} \leq v_j$$

$$\sum_i \sum_j v_i \geq x_{ij} + x_{ji}$$

Teniendo esto, es posible modelar el problema usando PuLP.

#### 4.6. Definición del modelo genético

Definimos métodos y representación utilizada para este este modelo. Inicialmente, tenemos que solucionar el problema según el camino de puntos que vamos a recorrer, para ello usamos una tupla que contiene los enteros del 1 al  $N$  donde  $N$  es el número de nodos de nuestro grafo. Dada la representación elegida y la realidad del problema, existen las siguientes restricciones para cada tupla:

- La tupla posee una cantidad y deben ser números no repetidos.

- Los datos tiene un sentido, los individuos que no cumplan con la representación de un camino dentro del grafo no pueden existir.

Para concluir, usamos una tupla que representa el índice de cada nodo.

#### 4.6.1. Función Fitness

La función objetivo es de un problema de minimización del esfuerzo de moverse de punto a punto según el peso planteado en la ecuación 4.8. Por lo que podemos decir que la función evaluará la siguiente suma:

$$\text{fitness} = \sum_i^{N-1} \text{Edge}(\text{solution}[i], \text{solution}[i + 1]) \quad (4.12)$$

#### 4.6.2. Operadores

##### 4.6.2.1. Selección

: Usamos la selección por proporción, es decir, que cada individuo tiene una probabilidad según su valor de fitness de ser elegido, dado por el cociente entre el valor del fitness del individuo sobre el total de la población (ecuación 4.13)

$$P_i = \frac{\text{fitness}(i)}{\sum_j^N \text{fitness}(j)} \quad (4.13)$$

##### 4.6.2.2. Recombinación

Se utilizo una versión modificada del operador de *Cruzamiento Basado en la Posición* (Position Based Crossover - PBX) donde se realizan los siguientes pasos:

- Seleccionar aleatoriamente varias posiciones del padre 1.
- Generar parcialmente el hijo 1, copiando los alelos de las posiciones elegidas del padre 1.
- Marcar los alelos del padre 2 que ya fueron seleccionados en el padre 1.
- Desde el inicio del padre 2, seleccionar secuencialmente el siguiente alelo que no haya sido marcado, y ponerlo en la primera posición libre del hijo 1, desde el comienzo.
- En caso el camino no sea consistente para el grafo, volver a repetir.

#### 4.6.2.3. Mutación

Se utilizó la **Mutación por Intercambio** (Exchange Mutation - EM). Es un operador muy simple de implementar, que sortea aleatoriamente dos posiciones en la solución y las intercambia. De ser necesario, en caso exista inconsistencia en el intercambio volvemos a intentar.

## 4.7. Definición del modelo de aprendizaje por refuerzo

El modelo a trabajar estará basado en una red profunda de Q-Learning, donde el entorno será una matriz de adyacencia de un *grafo dirigido pick-drop* de la ecuación 4.10, para ello debemos definir la estructura de dicho grafo de entrada.

La tarea del modelo será obtener la ruta de estos 3 pasajeros que poseen distancias relativas entre sí; sabemos que cada conexión entre estos puntos de pasajeros posee un valor ponderado según el valor que implique moverse de un punto a otros, por lo que se buscará minimizar este camino. La tarea del modelo con respecto a la matriz, tendrá únicamente las siguientes reglas:

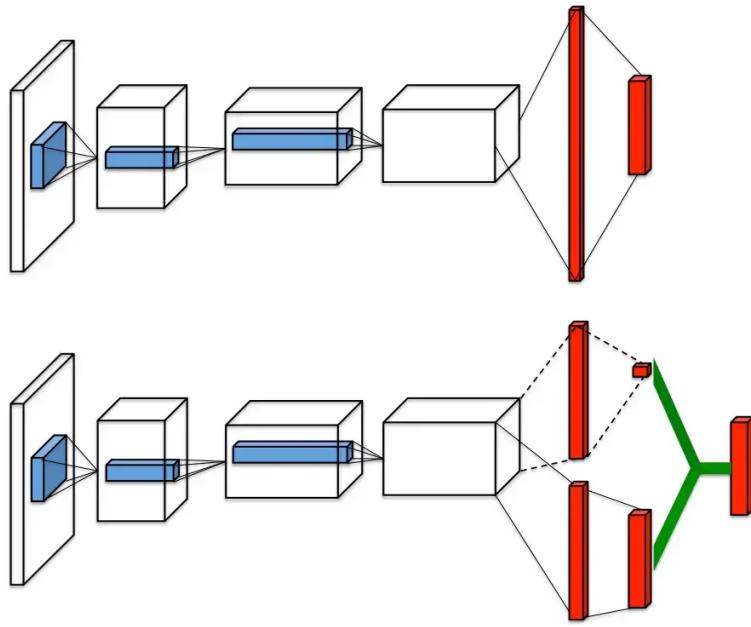
- Los puntos de recojo (**pickup**) están *totalmente conectados*, se permite dirigirnos a algún otro punto de recojo al igual que ir a cualquier punto de despacho.
- Los puntos de despacho (**dropoff**) están *parcialmente conectados*, esto se debe a que para cada punto de despacho no les será posible dirigirse a un punto de recojo.
- Solo existe un recorrido por cada punto, por lo que apenas se pasa por un punto, los nodos que conectan a este quedan eliminados en la siguiente movimiento.

### 4.7.1. Dueling DQN

A continuación definiremos cada uno de los parámetros presentes en nuestro modelo:

- Estado (S): En el problema de viajes compartidos, el estado incluye información como la posición actual del vehículo, el número de pasajeros a bordo, el destino deseado de los pasajeros y el valor actual recorrido.
- Acciones (A): El modelo tiene la decisión de recoger a un pasajero, dejar a un pasajero o elegir otro vehículo.
- Recompensa (R): La recompensa es una minimización de la cantidad de viaje recorrido.

Una arquitectura Dueling DQN a diferencia de la arquitectura DQN clásico, separa el proceso de estimación de la función de valor de estado en dos ramas paralelas: una para estimar la valoración de cada acción  $V(s)$ , y otra para estimar el valor promedio del estado al cual lo entenderemos como la ventaja  $A(s, a)$ .



**Figura 4.8:** Ilustración comparativa entre una arquitectura DQN clásica y una Dueling DQN. (Wang et al., 2015)

Para poder considerar la ventaja y la estimación de  $V$ , usaremos lo sugerido en Wang et al., 2015, ya que originalmente usamos una diferencia de la ventaja actual con la mejor, lo que lo hace “imperceptible” la ventaja de una acción sobre otra haciendo una convergencia más lenta.

Entonces, el valor  $Q(s, a)$  sera la valoración de la acción  $a$  con la diferencia entre la ventaja de acción ( $A(s, a)$ ) y la media de la ventaja todas las demás acciones, dando como resultado:

$$Q(s, a) = V(s) + (A(s, a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'))$$

## 4.8. Estructura del modelo de aprendizaje por refuerzo

El modelo utilizado, es un derivado de la arquitectura DQN (3.6), por lo que necesitaremos espacios de memoria para almacenar información temporal de estados y acciones.

En este modelo usamos una red Dueling DQN (Q-Net) que recibe como entrada las coordenadas y la matriz de adyacencia, retornando un arreglo del posible nodo a dirigirnos. Mientras que el

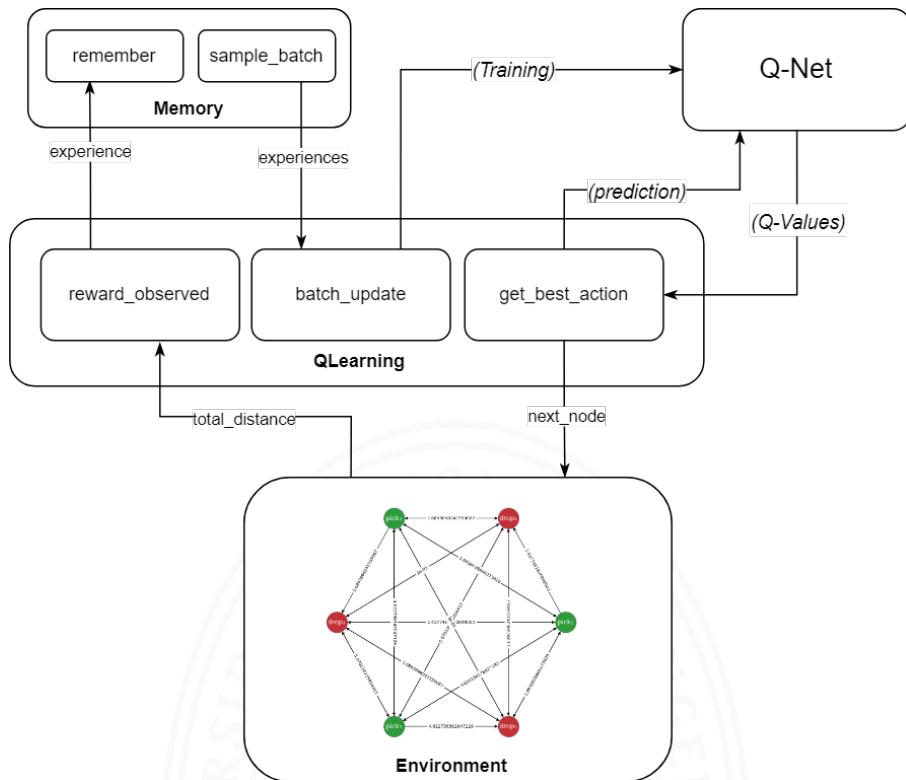


Figura 4.9: Ilustración de la estructura del modelo.

entorno nos envía la distancia total que será útil para realizar la recompensa relativa en cada estado, donde dicho valor es el valor a minimizar. Figura 4.9.

Además, se guarda la experiencia del modelo en memoria, como una forma de guardar el aprendizaje del modelo, y generar archivos de las mejores experiencias.

## 4.9. Red multi-modelo de precios

Se hizo uso de una red totalmente conectada, el cual tiene como función principal, retornar el precio adecuado según un conjunto de parámetros.

Esta red tiene un entrenamiento distinto, ya que a diferencia de los modelos antes propuestos, este tiene información de entrada y salida. Por lo que procedemos a describir según la figura 4.10 su comportamiento de la siguiente manera:

- **Capa de entrada:** La capa de entrada tendría tres neuronas, una para cada parámetro de entrada (distancia, tiempo y zona). Estas neuronas recibirían los valores de distancia en millas, tiempo estimado, recorrido y la cantidad de áreas que cruza (usando lo descrito en

la ecuación 4.6) y los pasarían a la siguiente capa.

- **Capa oculta:** La capa oculta, buscamos que no compleje la velocidad de ejecución del proceso, por lo que usamos una única capa con la cantidad de 4 neuronas en la capa oculta que usan una función de activación ReLU.
- **Capa de salida:** La capa de salida tendría una única neurona que recibiría los valores de la capa oculta para producir el precio como salida.
- **Entrenamiento:** La red neuronal se entrenaría con un conjunto de datos de la base de datos previamente mencionada. El objetivo del entrenamiento sería ajustar los pesos y los sesgos para la salida de precios más precisa posible dada las entradas.

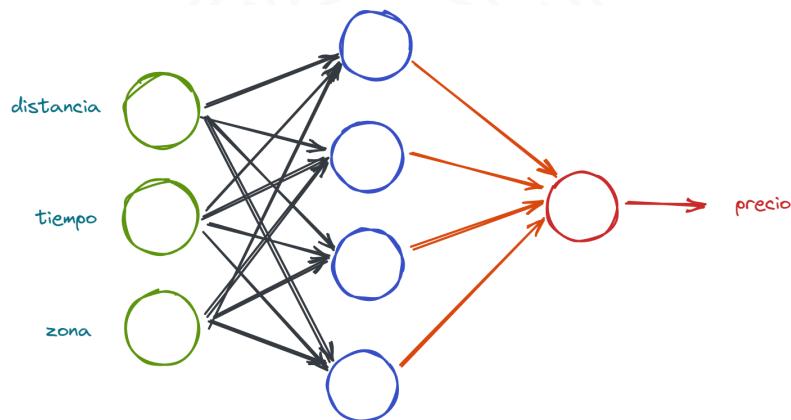


Figura 4.10: Ilustración de la estructura de la red neuronal.

Ahora bien, esta red neuronal está creada para una vez pre-entrenada ser usada para los distintos modelos antes expuestos, tal como se describe en la figura 4.11, se puede ver que cada modelo retorna una distancia distinta; sin embargo, usamos el tiempo y zona como datos extraídos de forma constante; como resultado, obtendremos 3 precios distintos, los cuales usaremos para nuestras métricas.

## 4.10. Flujo conceptual

Debido a la creación de un modelo capaz de trabajar con un conjunto de datos tan grande, el proyecto debía ser capaz de soportar una gran cantidad de épocas, en el menor tiempo posible, para esto, se definió un flujo de conexiones, que se puede ver representado en la figura 4.12.

El proyecto se conecta con una base de datos de PostgreSQL mediante un puerto abierto en el contenedor de Docker que lo tiene, para realizar las consultas necesarias y obtener los valores en

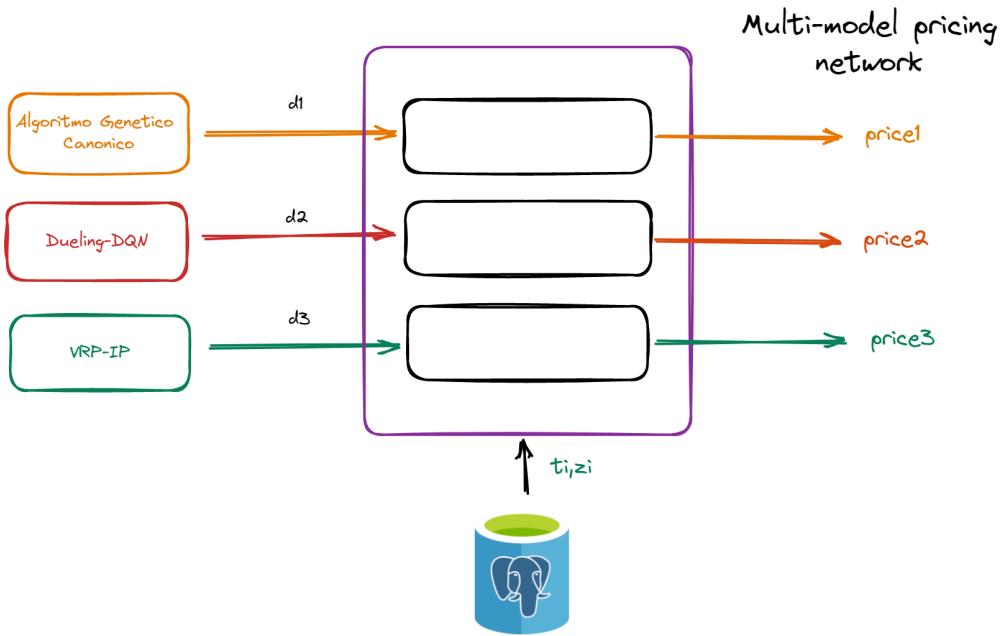


Figura 4.11: Ilustración del flujo de la red neuronal de precios conectado a los modelos comparativos.

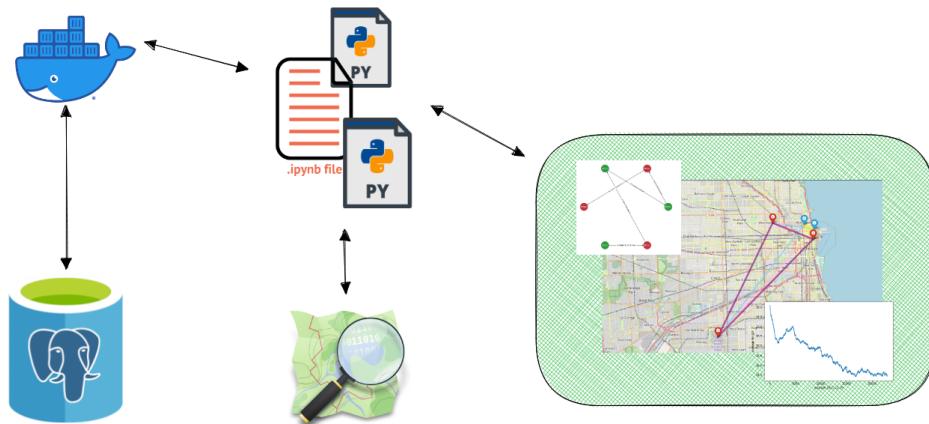


Figura 4.12: Ilustración del flujo de conexión del proyecto.

formato geoespacial, donde luego de entrenar los modelos necesarios consultamos la información de la zona en tiempo real mediante la API de OpenStreetMap, para finalmente mediante librerías de Python visualizar las respuestas.

Esta haría más accesible para el modelo seleccionar 3 puntos de recojo y 3 puntos de entrega, centrándonos únicamente al costo computacional del modelo.

# Resultados y Discusiones

## 5.1. Métricas

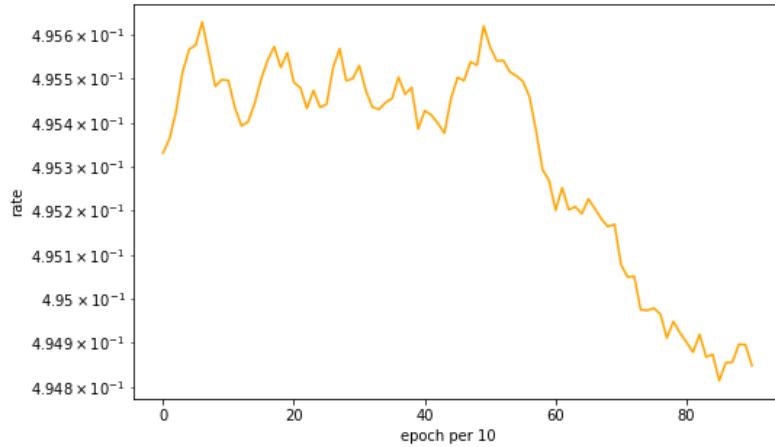
Se usaron 3 métricas generales para los 3 modelos de uso, de forma de encontrar las ventajas y desventajas del modelo de aprendizaje por refuerzo.

- **Métrica por varianza:** Como se vió en la figura 4.11 cada modelo tiene, idealmente, una distancia generada distinta, por lo tanto un precio distinto; esta métrica busca ver cuanto sesgo hay entre los modelos planteados.
- **Métrica por tiempos:** El tiempo que demora un modelo y otro en resolver el problema, guardando el tiempo que demora realizar cada uno de estos modelos.
- **Métrica por diversidad:** Se debe validar que los modelos pueden y aceptan diferentes casos, es decir, ser capaz de resolver eficientemente más de 1 solo grafo sin volver a realizar algún entrenamiento.

## 5.2. Comparación por modelo

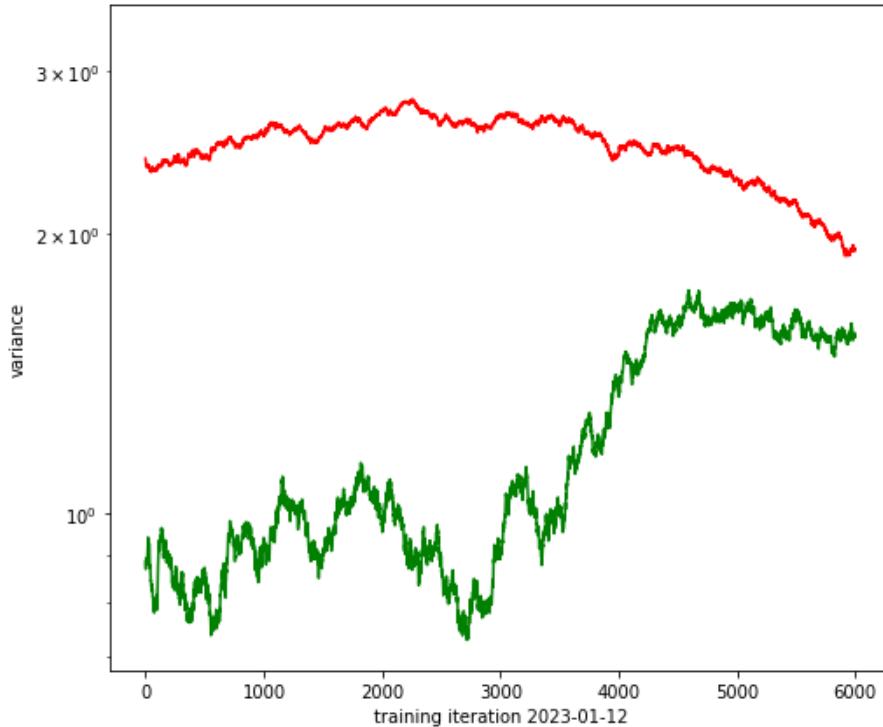
Para el uso de nuestras métricas hicimos un gráfico de resultados realizando la diferencia entre los precios de los modelos, sabiendo que la red neuronal fue previamente entrenada con aproximado de 1000 precios, como se ve en la gráfica 5.1, el cual no aumentamos la cantidad de pruebas para evitar el sobre-ajuste. Para comparar los modelos, nos basamos en la diferencia generada por únicamente el modelo Dueling-DQN. En la figura 5.2 podemos ver la diferencia de precios respecto al modelo de aprendizaje por refuerzo; de rojo, el modelo Dueling-DQN vs una variante VRP usando LP, mientras que, de verde, el algoritmo genético canónico vs Dueling-DQN.

Se puede ver mayor ruido oscilante de lado del modelo evolutivo a diferencia del modelo matemático. Esto demuestra la precisión del modelo matemático sobre la aproximación heurística. Ahora bien, conforme al numero de iteraciones avanza, la diferencia de precios logra



**Figura 5.1:** Gráfica de loss vs epoch por cada 10 iteraciones.

un balance destacando algo modelo Dueling-DQN para adaptarse al problema planteado.



**Figura 5.2:** Gráfica comparativo entre la diferencia de precios. De rojo, Dueling-DQN vs la variante VRP usando LP y de verde, algoritmo genético canónico vs el Dueling-DQN

Podemos ver que en el cuadro 5.1 la lista de tiempo de ejecución según 3 tipos, (1) tiempo de demora en converger a un resultado ideal, (2) tiempo en que demora entrenar el modelo, (3) tiempo en que demora resolver el problema.

Modelos	Tiempo de Convergencia	Tiempo Entrenamiento	Tiempo de Ejecución
Dueling-DQN	~7.4min	~32.2min	~0.9min
Genetic Algoritm	~10.2min	~22.3min	~0.3min
Variant VRP LP	~0min	~0min	~1.7min

Cuadro 5.1: Comparación de tiempo de ejecución de cada modelo.

De esto, vemos que el aprendizaje por refuerzo nos ofrece un consumo más alto para resolver el problema durante el entrenamiento, mientras que su tiempo de convergencia es menor al genético, esto podría deberse a que el modelo genético no generaliza soluciones para todos los grafos presentes; finalmente, observamos que el modelo de programación lineal no tiene un tiempo de convergencia ni entrenamiento que darnos, pero sí un tiempo de ejecución un poco más alto respecto a los demás.

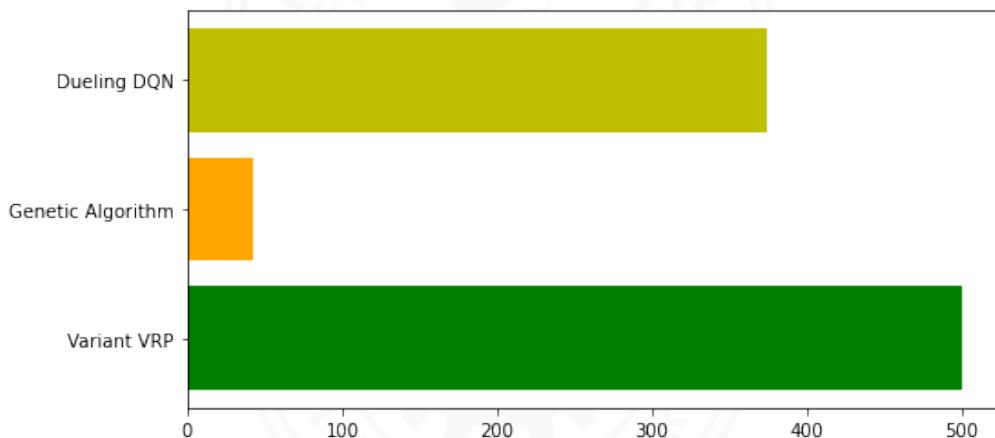


Figura 5.3: Barra comparativa de pruebas post aprendizaje.

En la figura 5.3 vemos la última métrica, métrica por diversidad, que es una lista de 500 casos de problemas de viajes compartidos, donde el modelo matemático acierta en cada uno, y que el modelo de Dueling DQN resuelve bien el 74.8 % de los casos y el algoritmo genético resuelve solo el 8.4 % de los casos debido a que la solución aparentemente no está bien generalizada.

### 5.3. Observaciones según el modelo

Con todo esto ya expuesto pudimos ver los siguientes observaciones.

- El modelo tiene mejor rendimiento que los modelos como DQN y Double DQN, debido a que no evalúa todas las acciones posibles respecto al estado actual bloqueando caminos con mejores recompensas a largo plazo. Esto se pudo ver gracias a que Dueling DQN divide en 2 partes: para calcular el valor esperado de cada estado y para calcular el mejor valor promedio según cada acción. Intentando siempre buscar el mejor movimiento del vehículo.
- La representación de cada individuo del modelo genético tenía fallas al generar nuevos hijos; debido a que, no cualquier hijo puede ser apto según las reglas del problema de viajes compartidos, acotando así el conjunto solución, teniendo que validar que la configuración de la cada individuo sea un camino permitido en el grafo, aumentando así el tiempo de ejecuciones más de lo debido.
- El modelo matemático basado en restricciones no tiene problemas en resolver el problema, sin embargo, no es posible inferir ni adaptarse a más problemas del que se plantea fuera de las reglas ya antes mencionadas.

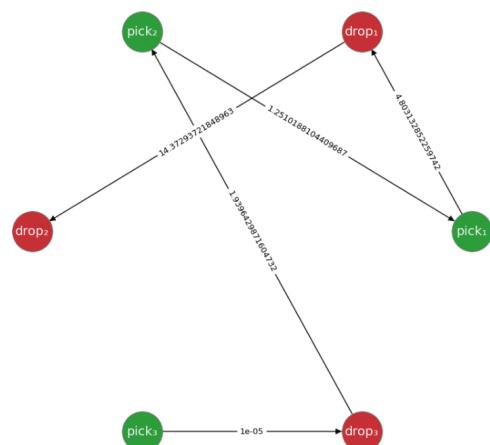
### 5.4. Gráficas

Python nos permite tener finalmente resultados visuales mediante la Api de OpenStreetMap, los cuales consumimos para poder reconocer el mapa y las rutas posibles para este. Por esto mismo generamos distintas gráficas según lo que necesitamos para entender el movimiento del vehículo y estas son:

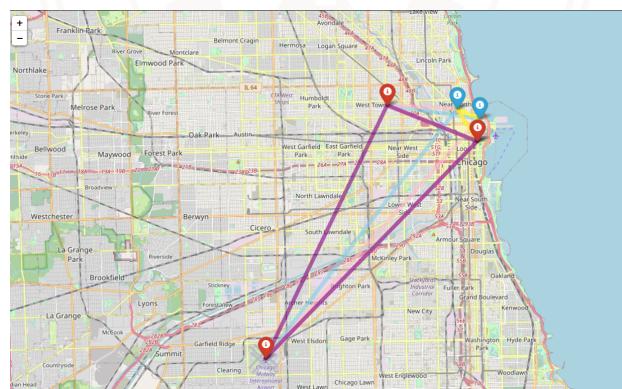
- Según el orden de los puntos, se traza una sola ruta.
- Según el orden de los nodos del grafo, se conectan los nodos.
- Según el mapa, se ve la distancia real de cada punto de recojo, respecto a su punto de destino.



(a) Según la ruta.



(b) Según el grafo.



(c) Según el mapa.

**Figura 5.4:** Gráficas de los puntos geográficos según los caminos tomados en diferentes representaciones.

# Conclusiones

## 6.1. Conclusiones

La investigación realizada ha demostrado que los modelos de aprendizaje por refuerzo profundo tienen mucho potencial para el campo de estimación de viajes compartidos; sabiendo que, sin una asignación del vehículo coherente, una ruta adecuada y un precio acorde los parámetros indicados, no sería posible el modelo de negocio de un servicio de viajes. De todo esto, podemos destacar que:

- Se diseñó una arquitectura completa que permite obtener solución al problema de viajes compartidos usando los conceptos de aprendizaje por refuerzo y los patrones de viajes compartidos. Empleando conceptos de asignación, enrutamiento y estimación de precios.
- Se formuló el grafo dirigido necesario que le permita a los diferentes modelos empleados en la investigación, sacarle el mejor provecho a la información presente en la base de datos de Chicago Taxi Trips.
- La arquitectura propuesta mostró la capacidad de adaptación de las redes neuronales conectadas a modelos completamente independientes, estimando precios ligeramente similares.
- Se pudo identificar y plantear los parámetros más comunes para resolver el problema de viajes compartidos y las condiciones para obtener su ruta ideal.

## 6.2. Trabajos futuros

- El servicio de viaje compartido es un libro abierto a diferentes propuestas de implementación que permitan mejorar la calidad del servicio de taxis.
- El uso de viajes compartidos de tipo dinámico, implicará resolver el problema de enrutamiento de vehículos con demanda dinámica (DVRP), una variante del problema VRP.
- El planteamiento del vehículo como agente individual nos ha ayudado a entender el comportamiento de los vehículos para realizar una ruta óptima; sin embargo, hace falta entender su comportamiento paralelo a varios agentes en el mismo entorno.

# Bibliografía

- Al-Abbasi, A. O., Ghosh, A., & Aggarwal, V. (2019). DeepPool: Distributed model-free algorithm for ride-sharing using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12), 4714-4727. <https://doi.org/10.1109/tits.2019.2931830>
- Almasan, P., Suárez-Varela, J., Badia-Sampera, A., Rusek, K., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case.
- Ambito. (2022). *Ahora Uber Permitirá compartir el viaje con otro usuario Y ahorrar hasta 30 %.* Consultado el 10 de mayo de 2022, desde <https://www.ambito.com/negocios/uber/ahora-permitira-compartir-el-viaje-otro-usuario-y-ahorrar-30-n5390468>
- Bayer, M. (2012). SQLAlchemy. En A. Brown & G. Wilson (Eds.), *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*. aosabook.org. <http://aosabook.org/en/sqlalchemy.html>
- Bei, X., & Zhang, S. (2018). Algorithms for Trip-Vehicle Assignment in Ride-Sharing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). <https://doi.org/10.1609/aaai.v32i1.11298>
- Bhogal, N. (2022). What is ridesharing / carpooling / vanpooling software? ridesharing vs. Ridehailing. <https://www.tripspark.com/blog/what-is-ridesharing-carpooling-vanpooling-software-definition>
- Boeing, G. (2017). OSMNX: New methods for acquiring, constructing, analyzing, and Visualizing Complex Street Networks. *Computers, Environment and Urban Systems*, 65, 126-139. <https://doi.org/10.1016/j.compenvurbsys.2017.05.004>
- Bond, S. (2020). *How uber and its riders are adapting to the new 'social'*. Consultado el 13 de mayo de 2022, desde <https://www.npr.org/2020/06/10/873499410/how-uber-and-its-riders-are-adapting-to-the-new-social>
- Braverman, A., Dai, J. G., Liu, X., & Ying, L. (2019). Empty-Car Routing in Ridesharing Systems. *Oper. Res.*, 67, 1437-1452.
- Cao, Y., Wang, S., & Li, J. (2021). The Optimization Model of Ride-Sharing Route for Ride Hailing Considering Both System Optimization and User Fairness. *Sustainability*, 13(2). <https://doi.org/10.3390/su13020902>

- CEPAL. (2022). Panel - Cities: Sustainable Mobility as an opportunity for a transformative recovery. <https://www.cepal.org/en/events/panel-cities-sustainable-mobility-opportunity-transformative-recovery>
- Chicago Data Portal. (2013). *Boundaries - Community Areas (current)* [Online]. <https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Community-Areas-current-/cauq-8yn6/data>  
Accedido: 2021-12-05
- Chicago Data Portal. (2019). *Update of Chicago Taxi Trips DataSet* [Online]. <http://dev.cityofchicago.org/open%20data/data%20portal/2019/07/01/taxi-dataset-relaunch.html>  
Accedido: 2021-12-05
- Chicago Portal. (2021). Taxi Trips [Online]. <https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew>  
Accedido: 2021-12-29
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Mit Press.
- Cristóbal, E. (2021). This Is The Future Of Ridesharing: What You Need To Know About Lyft And Uber Now [Online]. <https://www.forbes.com/sites/christopherelliott/2021/04/17/this-is-the-future-of-ridesharing-what-you-need-to-know-about-lyft-and-uber-now/?sh=5b4d81686916>  
Accedido: 2021-12-29
- Dailey, D., Loseff, D., & Meyers, D. (1999). Seattle smart traveler: dynamic ridematching on the World Wide Web. *Transportation Research Part C: Emerging Technologies*, 7(1), 17-32. [https://doi.org/https://doi.org/10.1016/S0968-090X\(99\)00007-8](https://doi.org/https://doi.org/10.1016/S0968-090X(99)00007-8)
- Dirac, G. A. (1952). Some Theorems on Abstract Graphs. *Proceedings of the London Mathematical Society*, s3-2(1), 69-81. <https://doi.org/https://doi.org/10.1112/plms/s3-2.1.69>
- Docker. (2022). Docker About. <https://www.docker.com/why-docker/>
- Elmaghraby, S. E. (2003). Operations research. *Encyclopedia of Physical Science and Technology*, 193-218. <https://doi.org/10.1016/b0-12-227410-5/00513-5>
- Fielbaum, A., Kronmueller, M., & Alonso-Mora, J. (2021). Anticipatory routing methods for an on-demand ridepooling mobility system. *Transportation*. <https://doi.org/10.1007/s11116-021-10232-1>
- Group PostgreSQL Global Development. (2022). <https://www.postgresql.org/about/>
- Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using NetworkX* (inf. téc.). Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

- Haliem, M., Mani, G., Aggarwal, V., & Bhargava, B. (2021). A Distributed Model-Free Ride-Sharing Approach for Joint Matching, Pricing, and Dispatching Using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 1-12. <https://doi.org/10.1109/TITS.2021.3096537>
- Hugh-jones, M. (1997). Beyond mapping: Concepts algorithms, and issues in GIS. *Preventive Veterinary Medicine*, 29, 345-347.
- Iain Dunnin, Stuart Mitchell & Michael O'Sullivan. (2011). PuLP: A Linear Programming Toolkit for Python. *Department of Engineering Science The University of Auckland*. <https://optimization-online.org/?p=11731>
- IBM Cloud Education. (2022). What is Docker? <https://www.ibm.com/cloud/learn/docker#toc-how-contai-5UTUfWRp>
- Instituto Nacional de Estadística e Informática. (2020). *Parque automotor en circulación a nivel nacional, según departamento (2011-2019)* [Online]. [https://www.inei.gob.pe/media/MenuRecursivo/indices\\_tematicos/cap20025\\_3.xlsx](https://www.inei.gob.pe/media/MenuRecursivo/indices_tematicos/cap20025_3.xlsx)  
Accedido: 2021-10-17
- Jordahl, K. (2014). GeoPandas: Python tools for geographic data. URL: <https://github.com/geopandas/geopandas>.
- Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85-103. [https://doi.org/10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9)
- Li, S., Li, M., & Lee, V. C. S. (2020). Trip-Vehicle Assignment Algorithms for Ride-Sharing. En W. Wu & Z. Zhang (Eds.), *Combinatorial Optimization and Applications* (pp. 681-696). Springer International Publishing.
- Lima Cómo Vamos. (2021, 25 de septiembre). *Informe urbano de percepción ciudadana en Lima y Callao 2021*. Consultado el 22 de septiembre de 2022, desde <http://www.limacomovamos.org/wp-content/uploads/2021/12/EncuestaLCV2021.pdf>
- Lima Cómo Vamos. (2022). Informe urbano de percepción ciudadana en Lima y Callao 2022. Consultado el 2 de noviembre de 2022, desde <http://www.limacomovamos.org/wp-content/uploads/2022/11/EncuestaLCV2022.pdf>
- Lyft. (2019). *Introducing Shared Saver, Our Most Affordable Ride* [Online]. <https://www.lyft.com/blog/posts/introducing-shared-saver>  
Accedido: 2021-10-19
- Merriam-Webster. (s.f.). SARS-CoV-2. Consultado el 4 de octubre de 2022, desde <https://www.merriam-webster.com/dictionary/SARS-CoV-2>
- Microsoft. (2020). Spatial Data (SQL Server) [Online]. <https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver15>  
Accedido: 2021-12-29

- Ministerio de Salud. (2020, 11 de marzo). DECRETO SUPREMO N° 008-2020-SA.
- Mnih, V., Kavukcuoglu, K., & Silver, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529-533. <https://doi.org/https://doi.org/10.1038/nature14236>
- Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers Industrial Engineering*, 79, 115-129. <https://doi.org/https://doi.org/10.1016/j.cie.2014.10.029>
- Morency, C. (2007). The ambivalence of ridesharing. *Transportation*, 34(2), 239-253. <https://doi.org/https://doi.org/10.1007/s11116-006-9101-9>
- Nechita, E., Crişan, G.-C., Obreja, S.-M., & Damian, C.-S. (2016). Intelligent Carpooling System. En K. Nakamatsu & R. Kountchev (Eds.), *New Approaches in Intelligent Control: Techniques, Methodologies and Applications* (pp. 43-72). Springer International Publishing. [https://doi.org/10.1007/978-3-319-32168-4\\_2](https://doi.org/10.1007/978-3-319-32168-4_2)
- Open Geospatial Consortium. (2022). About OGC. <https://www.ogc.org/about>
- OpenStreetMap contributors. (2017). Planet dump retrieved from <https://planet.osm.org>.
- Ore, O. (1960). Note on Hamilton Circuits. *The American Mathematical Monthly*, 67(1), 55-55. Consultado el 5 de agosto de 2022, desde <http://www.jstor.org/stable/2308928>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. En *Advances in Neural Information Processing Systems* 32 (pp. 8024-8035). Curran Associates, Inc. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- Perri, J. (2022, 19 de mayo). *The U.S. rideshare industry: Uber vs. lyft*. Consultado el 10 de junio de 2022, desde <https://secondmeasure.com/datapoints/rideshare-industry-overview/>
- PostGIS. (2022). About Postgis. <https://postgis.net/>
- Presidencia del Consejo de Ministros. (2022, 27 de octubre). DECRETO SUPREMO N° 130-2022-PCM.
- Psaraftis, H. N. (1980). A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*, 14(2), 130-154. <https://doi.org/10.1287/trsc.14.2.130>
- python-visualization. (2020, 28 de diciembre). Folium (Ver. 0.11.0). <https://github.com/python-visualization/folium/>

- Rahili, S., Riviere, B., Olivier, S., & Chung, S.-J. (2018). Optimal Routing for Autonomous Taxis using Distributed Reinforcement Learning. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 556-563. <https://doi.org/10.1109/ICDMW.2018.00087>
- Rana, P. (2020). *WSJ News Exclusive — Uber cuts 3,000 more jobs, shuts 45 offices in Coronavirus Crunch*. Consultado el 20 de mayo de 2022, desde <https://www.wsj.com/articles/uber-cuts-3-000-more-jobs-shuts-45-offices-in-coronavirus-crunch-11589814608>
- Schreieck, M., Safetli, H., Siddiqui, S. A., Pflügler, C., Wiesche, M., & Krcmar, H. (2016). A Matching Algorithm for Dynamic Ridesharing [Transforming Urban Mobility. mobil.TUM 2016. International Scientific Conference on Mobility and Transport. Conference Proceedings]. *Transportation Research Procedia*, 19, 272-285. <https://doi.org/https://doi.org/10.1016/j.trpro.2016.12.087>
- Shaheen, S., Cohen, A., & Bayen, A. (2018). The Benefits of Carpooling. *UC Berkeley: Transportation Sustainability Research Center*. <https://doi.org/https://doi.org/10.7922/G2DZ06GF>
- Sivanandam, S., & Deepa, S. N. (2008). *Introduction to genetic algorithms*. Springer Berlin Heidelberg.
- SohuNews. (2017). *Site carpooling aterrizó en shenzhen, para abrir la era del carpooling 2.0!* [Online]. [https://m.sohu.com/a/164857652\\_653684?\\_trans\\_=010004\\_pcwzy](https://m.sohu.com/a/164857652_653684?_trans_=010004_pcwzy)  
Accedido: 2021-10-19
- Song, J., Cho, Y. J., Kang, M. H., & Hwang, K. Y. (2020). An Application of Reinforced Learning-Based Dynamic Pricing for Improvement of Ridesharing Platform Service in Seoul. *Electronics*, 9(11), 1818. <https://doi.org/10.3390/electronics9111818>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. The MIT Press.
- The Nexus. (2021, 23 de agosto). *Transportation network companies*. Consultado el 11 de mayo de 2022, desde <https://www.urbanismnext.org/technologies/transportation-network-companies>
- The Wall Street Journal. (2022). *Uber Technologies Inc. Stock price & News - Wall Street*. Consultado el 10 de junio de 2022, desde <https://www.wsj.com/market-data/quotes/UBER?mod=chiclets>
- The Traveling Salesman Problem. (2008). En *Combinatorial Optimization: Theory and Algorithms* (pp. 527-562). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-71844-4\\_21](https://doi.org/10.1007/978-3-540-71844-4_21)
- Uber Blog. (2018). *Uber economy carpooling* [Online]. <https://www.uber.com/us/zh/ride/express-pool/>  
Accedido: 2021-10-19
- Uber Blog. (2022). ¿Aún no sabes qué es uber pool? Descubre Cómo funciona. Consultado el 10 de junio de 2022, desde <https://www.uber.com/es-PE/blog/que-es-uber-pool/>

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2015). Dueling Network Architectures for Deep Reinforcement Learning. <https://doi.org/10.48550/ARXIV.1511.06581>

Wolsey, L. A. (2021). *Integer Programming*. John Wiley & Sons, Inc.

Zhou, Z., & Roncoli, C. (2022). A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion. *Transportation Research Part C: Emerging Technologies*, 139, 103658. <https://doi.org/10.1016/j.trc.2022.103658>