



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

(INGENIERIA DE SISTEMAS) – (INVESTIGACION DE OPERACIONES)

Un sistema de simulación-optimización basado en agentes
para brindar servicios de emergencia en la red del STC

(MODALIDAD DE GRADUACIÓN: TESIS)

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

PRESENTA:

ING. ADRIÁN GUTIÉRREZ GÓMEZ

TUTORA PRINCIPAL

DRA. AIDA HUERTA BARRIENTOS. FACULTAD DE INGENIERÍA

Octubre de 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Rivera Colmenero José Antonio

Secretario: Dra. Aguilar Juárez Isabel Patricia

1 er. Vocal: Dra. Huerta Barrientos Aida

2 do. Vocal: M. I. Wellens Ann Godelieve

3 er. Vocal: M. I. Vera Morales Alma Elia

Lugar o lugares donde se realizó la tesis: CIUDAD UNIVERSITARIA, CDMX.

TUTORA DE TESIS:

DRA. AIDA HUERTA BARRIENTOS



FIRMA

Tabla de contenido

AGRADECIMIENTOS	11
RESUMEN	12
ABSTRACT	13
ABREVIACIONES, SIGLAS Y ACRÓNIMOS.....	14
INTRODUCCIÓN	15
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	17
1. LA PROBLEMÁTICA	17
1.1. EL CONTEXTO.....	17
1.2. EL PROBLEMA POR RESOLVER	28
1.3. LA SITUACIÓN DESEADA	29
1.4. EL OBJETIVO GENERAL.....	29
1.5. LOS OBJETIVOS ESPECÍFICOS.....	29
1.6. METODOLOGÍAS DE LA INVESTIGACIÓN.....	29
CAPÍTULO II: ESTADO DEL ARTE Y MARCO TEÓRICO:	31
2. ESTADO DEL ARTE	31
2.1. BÚSQUEDA LITERARIA	31
2.2. ANÁLISIS DE LA BÚSQUEDA LITERARIA	37
2.3. REVISIÓN LITERARIA	38
2.4. CRÍTICA DE LA LITERATURA	45
2.5. CONCLUSIÓN DE LA REVISIÓN DE LA LITERATURA	47
2.6. TIPOS DE MODELOS DE SIMULACIÓN EN INVESTIGACIÓN DE OPERACIONES	48
2.7. SIMULACIÓN DINÁMICA DE SISTEMAS	49
2.8. SIMULACIÓN DE EVENTOS DISCRETOS.....	50

2.9.	MODELACIÓN BASADA EN AGENTES	50
2.10.	METODOLOGÍA PARA EL DESARROLLO DEL MODELO DE SIMULACIÓN BASADO EN AGENTES.....	53
2.11.	ALGORITMO DIJKSTRA	55
2.12.	SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG).....	56
2.13.	QGIS.....	58
CAPÍTULO III: UN SISTEMA BASADO EN SIMULACIÓN-OPTIMIZACIÓN.....		59
3.	UN SISTEMA BASADO EN SIMULACIÓN - OPTIMIZACIÓN.....	59
3.1.	JUSTIFICACIÓN DE LA MODELACIÓN BASADA EN AGENTES (ABSM)	59
3.2.	JUSTIFICACIÓN DEL USO DEL SOFTWARE NETLOGO™	60
3.3.	SOFTWARE DE SIMULACIÓN NETLOGO™	61
3.4.	TRATAMIENTO DE MAPAS EN EL SOFTWARE QGIS™	62
3.5.	MODELO CONCEPTUAL	63
3.6.	ALGORITMO DEL MODELO	67
3.7.	DIAGRAMA DEL MOVIMIENTO DE TRENES	71
3.8.	MODELO COMPUTACIONAL	72
3.9.	OPTIMIZACIÓN EN EL MODELO DE SIMULACIÓN	86
3.10.	IMPLEMENTACIÓN DEL ALGORITMO DIJKSTRA EN NETLOGO™ 89	
3.11.	VERIFICACIÓN DEL MODELO.....	90
3.12.	VALIDACIÓN DEL MODELO.....	91
3.13.	REPLICACIÓN O REPRODUCCIÓN DEL MODELO	94
CAPÍTULO IV: EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS.....		96
4.	DISEÑO DE EXPERIMENTOS	96
4.1.	CARACTERÍSTICAS DE LA COMPUTADORA UTILIZADA.....	96
4.2.	BEHAVIORSPACE	97

4.3. CASOS ESPECIALES LÍNEA A y B	100
4.4. RESULTADOS DE LOS EXPERIMENTOS	102
4.5. ANALISIS DE LOS RESULTADOS DE LOS EXPERIMENTOS	104
4.6. RESULTADOS GENERALES	106
CONCLUSIONES.....	108
REFERENCIAS.....	112
ANEXO A: MODELACIÓN EN NETLOGO™	117
ANEXO B: EXPERIMENTO DE PROYECCIONES.....	127

Índice de Figuras

Figura 1.1 Mapa de la Zona Metropolitana del Valle de México.....	17
Figura 1.2 Distribución porcentual de viajes realizados un día entre semana por la población de 6 años y más en transporte público, según principales medio de transporte utilizado.	19
Figura 1.3 Afluencia anual por línea 2015. Fuente: Comisión de Derechos Humanos del Distrito Federal (2017).....	20
Figura 1.4 Estaciones de mayor afluencia promedio en día laborable.	21
Figura 1.5 Estructura orgánica general del STC.	22
Figura 1.6 Mapa de la red del STC.	23
Figura 1.7 Representación de un tren de nueve carros.	25
Figura 1.8 Representación de un tren de seis carros.....	25
Figura 1.9 Vista lateral de un carro.	26
Figura 1.10 Procesos en la revisión de la literatura.....	30
Figura 2.1 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelos de optimización, estaciones generales y de bomberos, y servicios de emergencias. 31	
Figura 2.2 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelos de optimización, últimos 10 años.	32
Figura 2.3 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta y estaciones de metro, 2012-2018.	33
Figura 2.4 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta, algoritmos y sistemas de información, 2012-2018.	33
Figura 2.5 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta y los métodos y técnicas de simulación, 2012-2018.	34

Figura 2.6 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta y estaciones de emergencia, 2012-2018.....	35
Figura 2.7 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta, metro y estaciones de emergencia, 2012-2018.....	35
Figura 2.8 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelación y simulación basada en agentes, 2012-2018.....	36
Figura 2.9 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelación y simulación basada en agentes, optimización y metro 2012-2018.	37
Figura 2.10 Metodología para el desarrollo del modelo de simulación basada en agentes.	53
Figura 2.11 Ejemplo de capas temáticas sobrepuestas en QGIS.	57
Figura 3.1 Capas de datos geográficos.....	63
Figura 3.2 Diagrama de flujo del modelo de simulación basado en agentes.	67
Figura 3.3 Diagrama de flujo para el movimiento de trenes.	71
Figura 3.4 Interfaz del modelo.....	72
Figura 3.5 Botones para configurar las condiciones iniciales del modelo.	73
Figura 3.6 “Switches” para habilitar las distintas líneas de la red del metro.....	73
Figura 3.7 Botones que permiten visualizar sub-modelos de cada línea de la red del metro.	74
Figura 3.8 Pantalla de visualización de los mapas y caja de reportes.	75
Figura 3.9 Sliders y switches de probabilidad y visualización de hospitales y estaciones de bomberos, choosers de recursos y botones de movimiento.	76
Figura 3.10 Contadores de incidentes para cada línea del metro y contadores de incidentes que involucren Ambulancias y Camiones de Bomberos.	77
Figura 3.11 Contadores de distancias media para cada línea hacia el hospital o estación de bomberos más cercana.....	77
Figura 3.12 Gráfica de incidentes por línea e histograma de rutas más cortas a hospitales por línea.	78

Figura 3.13 Histograma de rutas más cortas a estaciones de bomberos por línea.	78
Figura 3.14 Primera sección del código: "breeds".	79
Figura 3.15 Segunda sección del código: "propiedades".	80
Figura 3.16 Tercera sección del código: "separar trenes".	81
Figura 3.17 Cuarta sección del código: "cargando datos GIS".	81
Figura 3.18 Quinta sección del código: "eliminar duplicados".	82
Figura 3.19 Sexta sección del código: "Dibujar GIS con agentes".	82
Figura 3.20 Séptima sección del código: "to setup".	83
Figura 3.21 Octava sección del código: "Función Zoom".	84
Figura 3.22 Novena sección del código: "Movimiento de trenes".	85
Figura 3.23 Décima sección del código: "Función 'renueva' para el cambio de dirección".	85
Figura 3.24 Onceava sección del código: "Función 'llamar ambulancia' para generar la ruta más corta".	86
Figura 3.25 Ejemplo de optimización en el modelo de simulación basado en agentes.	87
Figura 3.26 Ejemplo de ruta más corta en el modelo de simulación-optimización basado en agentes para la línea 1, con recursos en "Low".	88
Figura 3.27 Algoritmo de Dijkstra en NetLogo TM	89
Figura 3.28 Primer espectro de validación.	92
Figura 3.29 Segundo espectro de validación.	93
Figura 4.1 Características del equipo utilizado para realizar los experimentos.	96
Figura 4.2 Ejemplo de experimento para el escenario I-A en la línea 1 del metro.	98
Figura 4.3 Línea A sin visualización de alcaldías pero con visualización de avenidas principales.	100
Figura 4.4 Línea B sin visualización de alcaldías pero con visualización de avenidas principales.	101

Figura 4.5 Tiempo promedio de ejecución del modelo para 1000 incidentes para cada escenario.	102
Figura 4.6 Promedio de incidentes relacionados con servicios de emergencia para 1000 incidentes totales en cada escenario. Fuente: Elaboración propia (2022) .	103
Figura 4.7 Promedio de Rutas Más Cortas del modelo para 1000 incidentes totales en cada escenario.	104
Figura A.1 Interfaz del software NetLogo™, modo de insertar Botones.....	122
Figura A.2 Interfaz del software NetLogo™, modo de insertar Deslizadores.	122
Figura A.3 Interfaz del software NetLogo™, modo de insertar Interruptores.....	123
Figura A.4 Interfaz del software NetLogo™, modo de insertar Seleccionadores.	123
Figura A.5 Interfaz del software NetLogo™, modo de insertar Entradas de Datos.	124
Figura A.6 Interfaz del software NetLogo™, modo de insertar Monitores.	124
Figura A.7 Interfaz del software NetLogo™, modo de insertar Gráficos.....	125
Figura A.8 Interfaz del software NetLogo™, modo de insertar Salidas de Datos.	125
Figura A.9 Interfaz del software NetLogo™, modo de insertar Notas de Texto...	126

Índice de Tablas

Tabla 1.1 Población de 6 años y más que realizó viajes en un día entre semana, por tipo y modo de transporte utilizando en el menos uno de los tramos de su(s) viaje(s) según sexo.	19
Tabla 1.2 Número de trenes por línea del STC.	24
Tabla 1.3 Capacidad de cada tren.	25
Tabla 1.4 Características generales del material rodante para los modelos neumáticos.	27
Tabla 2.1 Clasificación de los resultados de la búsqueda de la literatura.	38
Tabla 4.1 Ejemplo de tabla resultado para el experimento línea 1 I-B.	99
Tabla B.1 Experimento de proyecciones L1 escenario: IID.	127

AGRADECIMIENTOS

Doy gracias a mi familia por apoyarme, motivarme, cuidarme y ayudarme a crecer, así como de darme todas las oportunidades y herramientas que me han conducido hasta este punto, son el pilar sobre el cual me he levantado y desarrollado como humano.

Me siento sumamente agradecido de contar con mis amigos Huicho, Leo, Ro y Yuma, que me han apoyado, han estado para mí y conmigo estos años, hemos vivido muchas cosas juntos y mi vida no sería de esta forma sin ellos. También agradezco a Andy Arroyo por haber compartido parte de esta etapa conmigo.

Agradezco a la Dra. Aida Huerta, excelente profesionista a quien admiro demasiado, fui muy afortunado de conocer y más aún de trabajar con ella. Con su dirección, consejos y ánimos, logré realizar esta tesis y sin duda ha contribuido de manera considerable a mi crecimiento como profesionista.

Quiero agradecer a mis sinodales que dedicaron tiempo y esfuerzo para mejorar este trabajo.

Estaré eternamente agradecido con la Universidad Nacional Autónoma de México (UNAM), que ha sido mi segunda casa por una década. Me ha enseñado, me ha alimentado el cuerpo, mente y espíritu. Siempre estaré orgulloso de haber pertenecido a la UNAM y espero siempre llevar en alto el nombre de esta maravillosa institución.

Reconozco al personal del STC por proporcionar parte de la información para realizar este trabajo.

Quiero agradecer también, al departamento de psiquiatría de la UNAM, sin su ayuda definitivamente no habría llegado al posgrado.

Finalmente le doy las gracias a todas las personas que han contribuido a la realización de este trabajo en formas que podrían parecer insignificantes: como las personas que cultivaron los alimentos que consumí, las personas que me llevaron en el transporte a la facultad, etc. Yo al ser parte de un sistema, implica que este trabajo es un acto del sistema.

RESUMEN

En el Sistema de Transporte Colectivo (STC) de la Ciudad de México se presentan problemas complejos debido a averías en los trenes, por ejemplo la llanta ponchada, cuya atención requiere en ocasiones la asistencia de servicios de emergencia, quienes necesitan seleccionar la ruta más corta desde su ubicación hasta el lugar donde se presenta el incidente. Estudios en la literatura científica han examinado el uso de técnicas de optimización para resolver el problema de atención a emergencias en estaciones del metro aplicando el algoritmo de ruta más corta e integrando Sistemas de Información Geográfica (SIG). Sin embargo, las contribuciones para la solución de dicho problema son limitadas. El objetivo principal de este trabajo es diseñar, implementar y validar un sistema de simulación-optimización basado en agentes para brindar servicios de emergencia desde su origen hacia el lugar donde se presenta el incidente en la red del metro, mediante la ruta más corta establecida en avenidas principales de la Ciudad de México, utilizando el software NetLogo™. Primero, se presenta la revisión de la literatura. Después, se implementa el sistema de simulación-optimización integrado con SIG utilizando el software NetLogo™. Enseguida, se analizan los resultados de la simulación-optimización y se diseñan escenarios anticipatorios. Finalmente, se enuncian las conclusiones de las cuales destaca la creación de un indicador de cercanía de los servicios de emergencia a cualquier punto sobre las líneas del metro.

ABSTRACT

Complex problems arise in the Sistema de Transporte Colectivo (STC) of Mexico City due to breakdowns in trains, for example a flat tire, whose attention sometimes requires the assistance of emergency services, who need to select the shortest route from their location to the place where the incident occurs. Studies in the scientific literature have examined the use of optimization techniques to solve the emergency response problem in subway stations by applying the shortest route algorithm and integrating Geographic Information Systems (GIS). However, the contributions for the solution of this problem are limited. The main objective of this work is to design, implement and validate an agent-based simulation-optimization system to provide emergency services from their origin to the place where the incident occurs in the metro network, through the shortest route established in main avenues of Mexico City, using NetLogo™ software. First, the review of the literature is presented. Then, the simulation-optimization system integrated with GIS is implemented using the NetLogo™ software. Next, the results of the simulation-optimization are analyzed and anticipatory scenarios are designed. Finally, the conclusions are stated, from which the creation of an indicator of proximity of the emergency services to any point on the subway lines stands out.

ABREVIACIONES, SIGLAS Y ACRÓNIMOS

ABM: Agent Based Modeling (Modelación Basada en Agentes)

CDMX: Ciudad de México

DES: Discrete Event Simulation (Simulación de Eventos Discretos)

GIS: Geographic Information Systems (Sistemas de Información Geográfica o SIG)

RMC: Ruta Más Corta

Tic: Unidad de tiempo en NetLogo™ (en inglés Tick del sonido de un reloj)

SD: System Dynamics (Dinámica de Sistemas)

STC: Sistema de Transporte Colectivo

ZMVM: Zona Metropolitana del Valle de México

INTRODUCCIÓN

En la operación del Sistema de Transporte Colectivo (en adelante STC) de la Ciudad de México (en adelante CDMX), existen diversas situaciones que afectan de manera directa la calidad del servicio y a su vez repercuten claramente en la percepción del usuario, causando insatisfacción con el servicio. Averías en los trenes, lluvias y accidentados en las vías principales, son algunos ejemplos que generan retrasos en la circulación de los trenes.

Un aspecto que influye de manera crucial en la insatisfacción de la calidad del servicio que percibe el usuario, es el tiempo de atención a un incidente, siendo el tema de interés de este estudio el incidente denominado *cortocircuito por llanta ponchada*. El área de ingeniería del STC requiere determinar (la distancia máxima recorrida de los servicios de emergencia como ambulancias y bomberos desde su localización hasta cualquier punto de la red del metro en donde se presente el incidente *cortocircuito por llanta ponchada*, así como analizar escenarios de optimización. Lo anterior como parte de su sistema de operación y mantenimiento

Un gran número de estudios en la literatura científica ya han examinado el uso de las técnicas de optimización para resolver el problema de atención a emergencias en estaciones del metro aplicando el algoritmo de ruta más corta e integrando Sistemas de Información Geográfica (SIG). Sin embargo, las contribuciones basadas en técnicas de simulación-optimización y su integración con SIG para la solución de dicho problema son limitadas.

El objetivo principal de trabajo de investigación es diseñar, implementar y validar un sistema basado en simulación-optimización para brindar servicios de emergencia desde hospitales públicos y estaciones de bomberos hacia el lugar donde se presenta el incidente denominado *cortocircuito por llanta ponchada* en la red del metro de la Ciudad de México, utilizando el software NetLogoTM. Se escoge utilizar la modelación y simulación basada en agentes debido a cuatro razones: la primera es que se considera la movilidad de la CDMX como un sistema complejo del cual no se conoce su comportamiento agregado (requisito para utilizar dinámica de sistemas). La segunda razón, se identifican y consideran agentes heterogéneos y sus interacciones (trenes, hospitales y estaciones de bomberos). La tercera razón es que la modelación basada en agentes permite describir el comportamiento del sistema complejo en estudio de una manera natural y fácil de comprender. Finalmente, con la modelación basada en agentes se puede considerar el movimiento en el espacio (coordenadas XY) y tiempo, lo cual no se puede considerar con entidades (Simulación de Eventos Discretos) o procesos (Dinámica de Sistemas).

Este trabajo de investigación consta de cuatro capítulos. En el **Capítulo 1**, se describe con detalle la problemática que se presenta en el metro en relación con los incidentes que se generan a lo largo de su trayecto. Se detalla el problema a resolver. Se enuncian tanto el objetivo general que funciona como guía y eje

principal en el desarrollo de este trabajo de investigación, así como los objetivos específicos que complementan y soportan al objetivo general. Se concluye este capítulo con la caracterización de la metodología de investigación que se utiliza.

En el **Capítulo 2**, se revisa la literatura relacionada con los modelos de optimización que se utilizan en la actualidad y que estén relacionados con los temas de estaciones generales y de bomberos además contemplando servicios de emergencia como las ambulancias. También, se revisa la literatura relacionada con el uso de software de simulación en la solución del problema de optimización de ruta más corta, se hace énfasis en la simulación basada en agentes. Cada una de las líneas de búsqueda de la literatura relaciona el tema del metro con distintas metodologías como la simulación basada en agentes, la optimización o el modelo de ruta más corta. Además, se incluyen como marco teórico los tipos de modelos de simulación utilizados en la Investigación de Operaciones.

En el **Capítulo 3**, se describe brevemente el software NetLogo™ y se justifica su utilización en este trabajo de investigación. También, se justifica el uso de ABMS. Enseguida, se diseña e implementa un sistema basado en simulación-optimización integrado con SIG utilizando el software NetLogo™ para brindar servicios de emergencia a incidentes en el metro en el caso de la contingencia de *cortocircuito por llanta ponchada*, siguiendo la metodología para ABMS.

En el **Capítulo 4**, se analizan los resultados de la simulación-optimización basada en agentes y se diseñan escenarios anticipatorios descritos en el modelo conceptual del modelo en el **Capítulo 3**. Así mismo, se obtienen algunas conclusiones sobre los datos y resultados del modelo, para posteriormente dar sugerencias sobre posibles líneas de estudio.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.LA PROBLEMÁTICA

1.1. EL CONTEXTO

De acuerdo con el entonces Gobierno del Distrito Federal, la Secretaría de Desarrollo Social y el Gobierno del Estado de México, (1998), la Zona Metropolitana del Valle de México (ZMVM) se puede definir como el conjunto conformado por las dieciséis alcaldías de la CDMX, más cincuenta y nueve municipios del Estado de México y el municipio de Tizayuca en el Estado de Hidalgo. Estos elementos del conjunto urbano constantemente interactúan en todos los ámbitos de la vida económica de la capital, hecho fácilmente constatable en los flujos de viajes que se realizan a diario entre las alcaldías y municipios. En la figura 1.1 se presenta un mapa de la Zona Metropolitana del Valle de México (ZMVM).

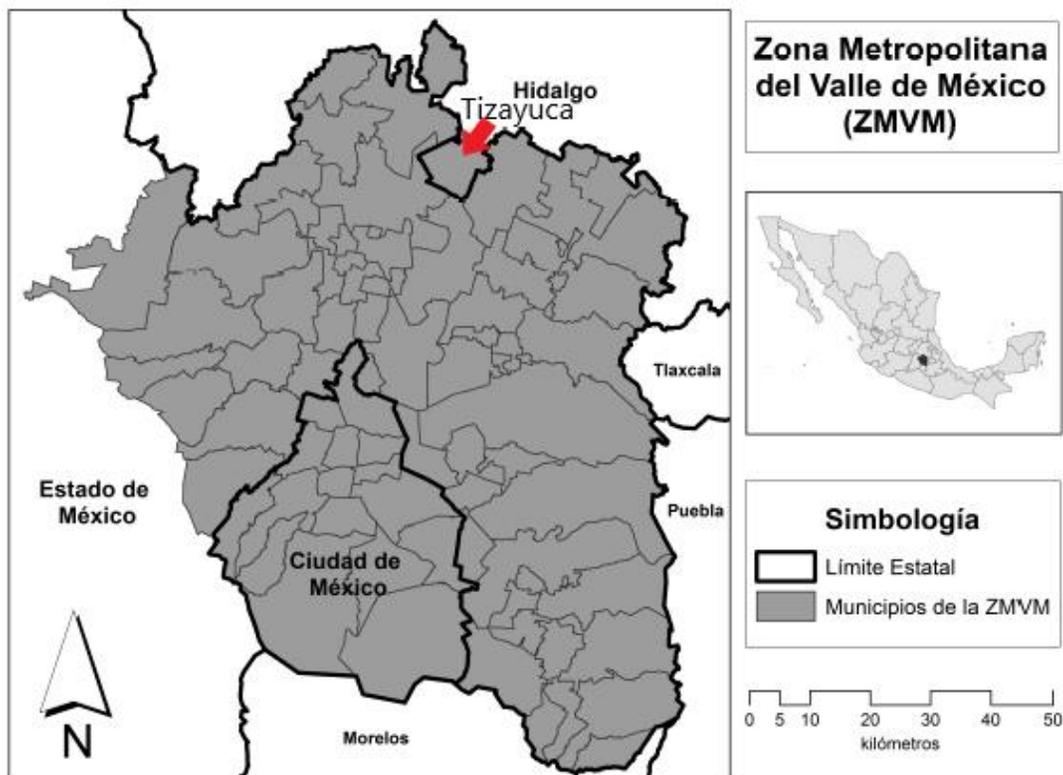


Figura 1.1 Mapa de la Zona Metropolitana del Valle de México.
Fuente: Salinas-Arreortua (2017).

En el plano económico la ZMVM genera el 29% del Producto Interno Bruto (PIB) del país y a su vez, el 73% del PIB de la ZMVM lo provee la CDMX; asimismo el 17% del PIB nacional lo aportan las alcaldías Miguel Hidalgo, Benito Juárez, Álvaro Obregón, Coyoacán y Cuauhtémoc. En términos geográficos, de acuerdo con datos del Censo de Población del 2020 (INEGI, 2021), 26,202,362 millones personas habitan en la ZMVM, considerando la población de la CDMX 9,209,944, representa un 35.14% de la población de la ZMVM y un 7.31% de la población total de México (126,014,024 habitantes), lo cual resulta sorprendente si se tiene en cuenta que la CDMX representa el 0.08% de la superficie total de la República Mexicana.

Debido a la importancia tanto económica como geográfica, la ZMVM está por encima de cualquier otra conurbación en el país en cuestiones de movilidad. Particularmente la Ciudad de México mantiene una relevancia significativa en los traslados de la región debido a la intensidad, frecuencia y volumen de movilidad que se presenta. La movilidad cotidiana de una población constituye un elemento de gran interés en el estudio de diferentes temas, útiles para la planeación de la infraestructura urbana y desplazamientos, así como para el análisis de los vínculos entre características sociodemográficas y movilidad habitual como indica el INEGI en su página de la encuesta origen-destino 2017 (INEGI, 2021). Esta encuesta fue levantada entre el periodo del 23 de enero al 3 de marzo del 2017, con el objetivo de conocer la movilidad de los habitantes de la ZMVM.

La Encuesta de Origen-Destino (EOD) considera una población de 19.38 millones de personas de 6 años o más en la ZMVM, de las cuales, poco más del 80% (15.63 millones) realiza al menos un viaje entre semana (INEGI, 2017). Esta proporción es del 84% en la CDMX y del 78% en los municipios conurbados. De la misma forma, de las 15.63 millones de personas que realizan al menos un viaje entre semana en la ZMVM, el 50.6% son mujeres y el 49.4 son hombres. El número de personas que realizan al menos un viaje los sábados en la población de 6 años y más se reduce al 53% (10.35 millones).

Se establecen dos grandes grupos de modo de transporte en la ZMVM de acuerdo con la encuesta origen-destino (INEGI, 2017), el transporte público y el transporte privado, de los cuales el transporte público representa cerca de la mitad del modo de transporte utilizado por la población. De la tabla 1.1, se reconoce la importancia que representa el metro dentro de la movilidad en la ZMVM, ya que de la población de 15.62 millones que realiza al menos un viaje en un día entre semana, 2.39 millones de personas (> 15%) lo hacen en metro. De forma similar, se producen 34.56 millones de viajes un día entre semana en la ZMVM (INEGI, 2017), de los cuales 11.15 son exclusivamente caminando, dejando 23.41 millones de viajes en transporte, de los cuales 15.57 millones son en transporte público y como se observa en la figura 1.2, el metro proporcionó 4.47 millones de viajes de los 15.57 millones (28.7%, lo que implica el segundo lugar).

Tabla 1.1 Población de 6 años y más que realizó viajes en un día entre semana, por tipo y modo de transporte utilizando en el menos uno de los tramos de su(s) viaje(s) según sexo.

Fuente: INEGI. Encuesta Origen - Destino en Hogares de la Zona Metropolitana del Valle de México (EOD) 2017.

Tipo y modo de transporte	Total		Sexo			
	Millones de personas ¹	%	Mujeres	%	Hombres	%
Zona metropolitana del Valle de México	15.62		7.91		7.72	
Transporte público ²	7.96	50.9	3.95	49.9	4.01	52
Colectivo	6.09	76.6	2.96	75.1	3.13	78
Taxi (calle, sitio o aplicación)	1.07	13.4	0.68	17.3	0.39	9.7
Metro	2.39	30	1.01	25.5	1.38	34.4
Metrobús o Mexibús	0.62	7.8	0.3	7.5	0.32	8.0
Otro transporte público	1.2	15.1	0.58	14.7	0.62	15.5
Transporte privado ³	3.48	22.3	1.44	18.2	2.04	26.4
Automóvil	3.15	90.5	1.32	91.6	1.83	89.8
Motocicleta	0.18	5.1	0.04	3	0.13	6.5
Transporte escolar	0.16	4.5	0.08	5.6	0.08	3.7
Transporte personal	0.02	0.6	0.01	0.7	0.01	0.6
Caminar en la calle	10.3	65.9	5.65	71.4	4.65	60.3
Bicicleta ⁴	0.34	2.2	0.09	1.1	0.26	3.3
Otro ⁵	0.02	0.2	0.01	0.1	0.02	0.2

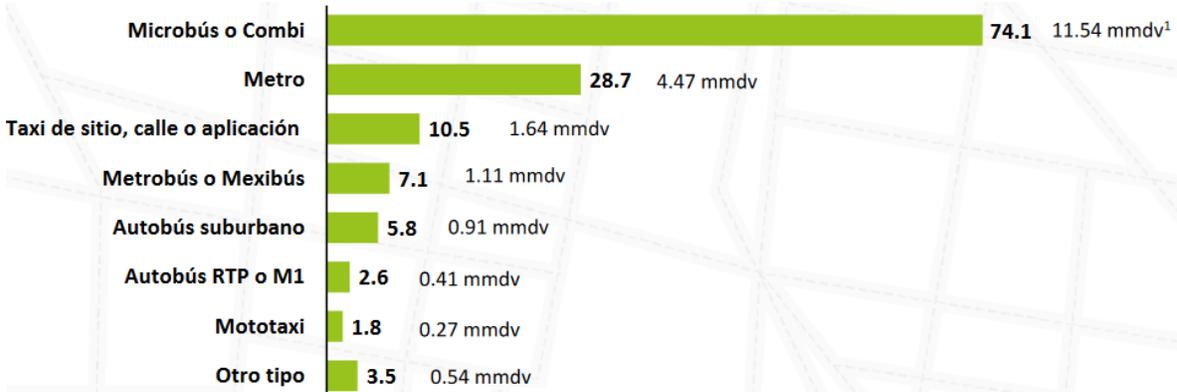
¹ La suma por modo de transporte utilizado, puede ser mayor que su total correspondiente, debido a que una persona puede emplear más de un modo de transporte

² El porcentaje de los transportes se obtiene con respecto al total de transporte público.

³ El porcentaje de los transportes se obtiene con respecto al total de transporte privado.

⁴ Debido a la forma en que se captó la bicicleta, no es posible si es transporte público o privado.

⁵ En la categoría "otro" se incluyeron trajinera y lancha, entre otros.



¹ mmdv= millones de viajes

Nota: Otro tipo considera tren suburbano, trolebús, tren ligero, bicitaxi y MEXICABLE.

La suma de la utilización de los medios de transporte es mayor que el 100%, porque en un viaje la persona puede utilizar más de un medio de transporte
Fuente: INEGI. Encuesta Origen - Destino en Hogares de la Zona Metropolitana del Valle de México (EOD) 2017.

Figura 1.2 Distribución porcentual de viajes realizados un día entre semana por la población de 6 años y más en transporte público, según principales medio de transporte utilizado.

Fuente: INEGI. Encuesta Origen - Destino en Hogares de la Zona Metropolitana del Valle de México (EOD) 2017.

El STC es un organismo público y descentralizado, cuyo objetivo es la operación de un tren rápido (metro), fue creado el 29 de abril de 1967, mediante su publicación en el Diario Oficial de la Federación. Se define como sistema por ser un conjunto de instalaciones fijas (redes y terminales), entidades de flujo como

vehículos, material rodante y un sistema de control que permiten movilizar eficientemente personas y bienes, para satisfacer necesidad de movilidad (STC, 2017). Los trenes del STC son movidos por energía eléctrica y la red cuenta con recorridos subterráneo, de superficie y elevado para dar movilidad principalmente a usuarios de la Ciudad de México y la Zona Metropolitana del Valle de México. El metro fue inaugurado el 4 de septiembre de 1969 con el primer tramo de la línea 1, entrando en operaciones normales el 5 de septiembre del 1969.

El STC es un ícono de la Ciudad de México, además es un servicio vital para la movilidad en la Zona Metropolitana del Valle de México (ZMVM), con las cifras indicadas en el *DIAGNÓSTICO SOBRE EL SERVICIO Y LAS INSTALACIONES DEL SISTEMA DE TRANSPORTE COLECTIVO 2013-2018* (STC, 2017). Durante el 2018 el metro transportó a 1,623 millones de personas al año, equivalente a 4.9 millones de personas en un día laborable y más de 8 millones de tramos viaje (segmento de viaje que se realiza a bordo de un vehículo de transporte) al día considerando transbordos.

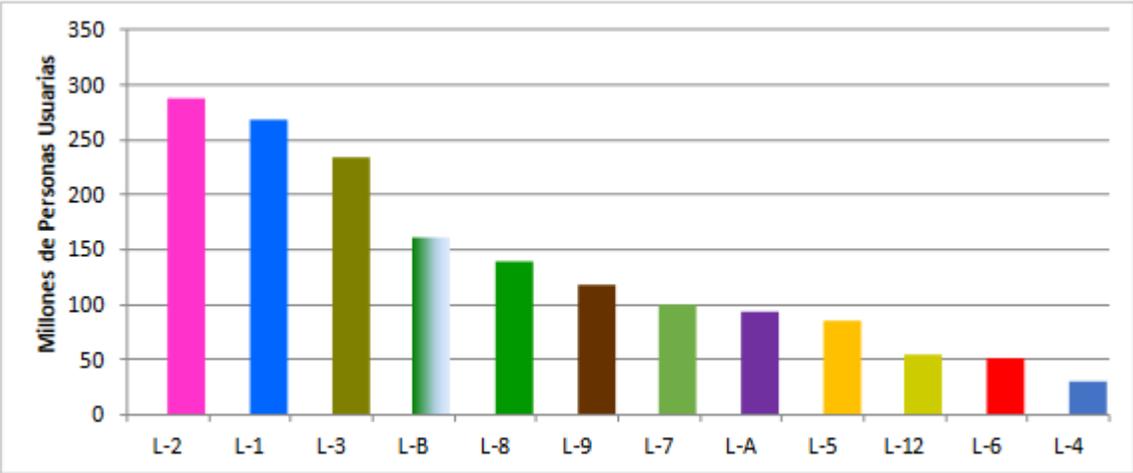


Figura 1.3 Afluencia anual por línea 2015.
Fuente: Comisión de Derechos Humanos del Distrito Federal (2017)

En la figura 1.3 se observa en orden la afluencia, en millones de personas usuarias, de cada línea del metro. Destacan la línea 1, 2 y 3 con más de 225 millones de usuarios. De la totalidad de su afluencia, aproximadamente el 12% son con acceso gratuito, esta afluencia se debe a que el servicio operado por el STC se ofrece los 365 días del año, con un horario en días laborables de 5:00 a 00:00 horas; los sábados de 6:00 a 0:00, y los domingos y días festivos de 7:00 a 00:00. De la figura 1.4, se observa que las estaciones con mayor afluencia de personas son: Indios Verdes (línea 3), Pantitlán (línea A), Cuatro Caminos (línea 2) y Constitución de 1917 (línea 8).

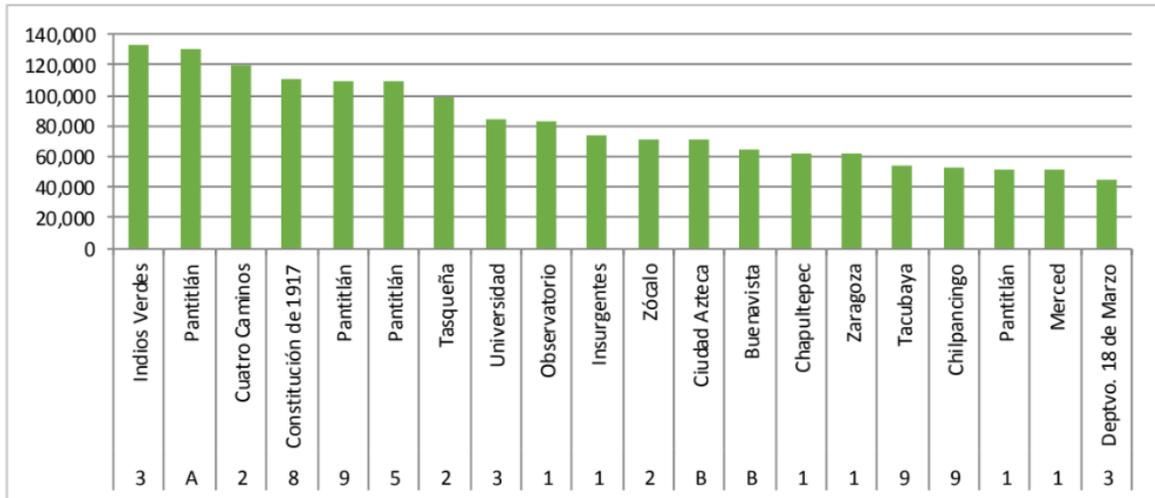


Figura 1.4 Estaciones de mayor afluencia promedio en día laborable.
Fuente: Comisión de Derechos Humanos del Distrito Federal (2017)

El Sistema de Transporte Colectivo es importante para la ZMVM y varios aspectos. En el aspecto económico es una de las empresas públicas más importantes del país con un valor de 239 mil millones de pesos de acuerdo con la CNDH (2017). En un enfoque cultural y social, parte de la misión y visión del STC es brindar de forma diaria una oferta cultural amplia y de calidad que reúne a algunos de los principales representantes de las artes escénicas, visuales y literarias del ámbito nacional e internacional, esto mediante su programa cultural, por el cual se realizan eventos de música, teatro, danza y literarios en sus 195 estaciones, asimismo cuenta con exposiciones en sus vitrinas de exhibición cultural.

De acuerdo con el Diagnóstico sobre el Servicio y las Instalaciones del Sistema de Transporte Colectivo 2013-2018 (STC, 2017), la estructura orgánica general del STC, figura 1.5, está compuesta por tres áreas:

- Subdirección general compuesta por:
 - Operación
 - Mantenimiento
 - Administración y Finanzas
- Dirección que responde a las áreas de subdirección general y dirección general
- Gerencia que responde al área de dirección y dirección general.

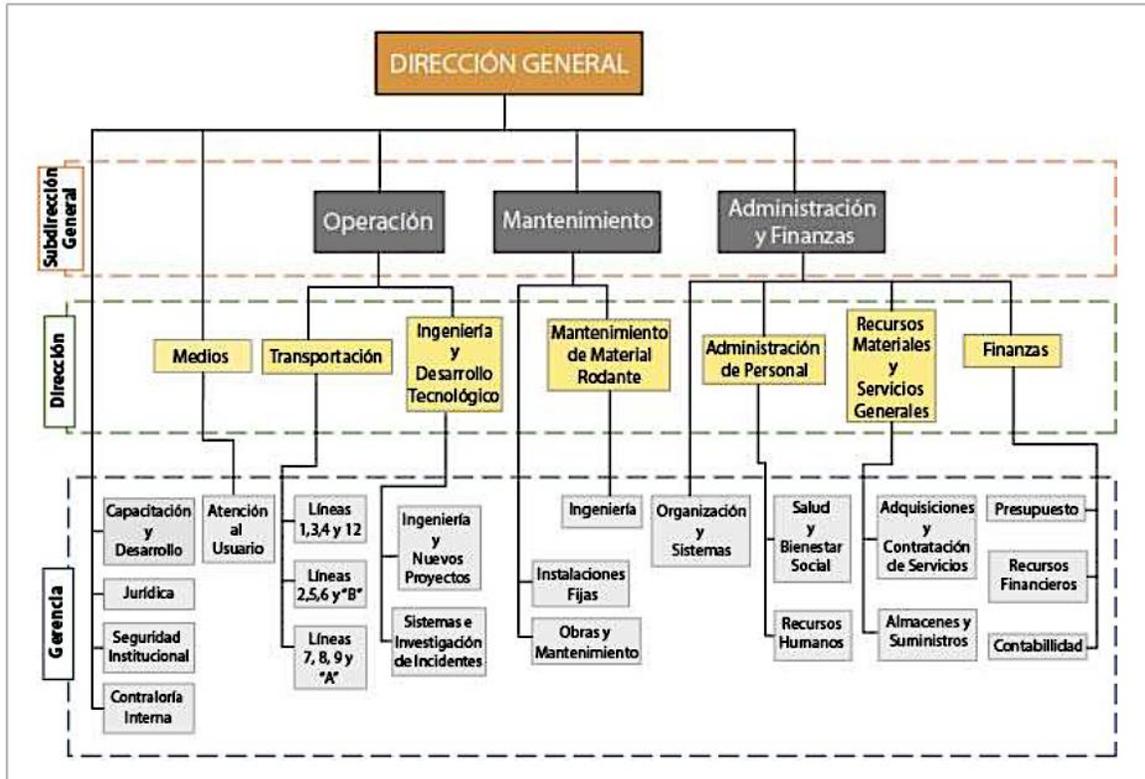


Figura 1.5 Estructura orgánica general del STC.
Fuente: Comisión de Derechos Humanos del Distrito Federal (2017).

El *DIAGNÓSTICO SOBRE EL SERVICIO Y LAS INSTALACIONES DEL SISTEMA DE TRANSPORTE COLECTIVO 2013-2018* (STC, 2017) hace una descripción de la red del metro, la cual se puede observar en la figura 1.6 y básicamente es radial.



Figura 1.6 Mapa de la red del STC.
Fuente: STC (2021).

De la red del metro en la figura 1.5, se observa que de sus 12 líneas integradas por 226 km de vías y 195 estaciones, de las cuales 44 son estaciones de correspondencia y 127 de paso, de la misma forma existen 12 estaciones terminales con correspondencia y 12 terminales sin correspondencia. El contar con un elevado número de correspondencias y estaciones terminales con correspondencia, permite agilizar el traslado de personas usuarias e integrar

trayectos en la CDMX y municipios conurbados del Estado de México. Por su tipo existen 115 estaciones subterráneas, 55 superficiales y 25 elevadas. Tres de las doce líneas constituyen una red troncal de alta captación, integrada por las líneas de mayor antigüedad que son la 1, 2 y 3 las cuales concentran el 49% de la afluencia promedio en día laborable. Con respecto al trazo restante de la red, destacan las líneas radiales que ofrecen una cobertura al territorio mexiquense: línea “B” al nororiente, desde Ciudad Azteca a Buenavista y línea “A” al oriente, desde Los Reyes-La Paz a Pantitlán. Por otro lado, el trazo de las líneas 5, 6 y 7 forman los arcos norte, oriente y poniente de un circuito de metro que se aproxima al trazo del Circuito Interior. La línea 8 conecta el suroriente con el centro de la ciudad y la línea 9 tiene un recorrido oriente-poniente con transbordos directos a las líneas 1, 2, 3, 4, 5, 7, 8 y “A” y la línea 12 con un recorrido que va de oriente-poniente, cruzando transversalmente la Ciudad de México.

Para ofrecer el servicio de traslado entre los distintos puntos de destino de la Ciudad de México y Zona Metropolitana, el STC cuenta con un parque vehicular de 394 trenes, de los cuales, 331 son de rodadura neumática, integrados por 302 trenes de 9 carros y 29 de 6 carros, así como 63 de rodadura férrea, integrados por 22 trenes de 9 carros y 22 de 6 carros, así como 30 trenes de 7 carros y 11 de 6 carros (STC, 2022).

La red del sistema de Transporte Colectivo tiene un total de 394 trenes asignados (330 neumáticos y 63 férreos) como se observa en la tabla 1.2, la cantidad de trenes por línea puede variar por una redistribución del parque vehicular en función de la necesidad de operación del servicio.

Tabla 1.2 Número de trenes por línea del STC.
Fuente: STC (2022).

Línea	1	2	3	4	5	6	7	8	9	A	B	12	Total
Trenes	50	41	54	14	25	15	32	30	34	33	36	30	393

El parque vehicular de la STC está conformado en su mayor parte por modelos neumáticos con la excepción de 4 modelos férreos. En los modelos se distinguen dos tipos de tecnología en su fabricación: 1) árbol de levas, 2) semiconductores y controles electrónicos. El 5% cuenta con sistema de tracción-frenado del tipo electromecánico (árbol de levas), mientras que el resto cuenta con un sistema de tracción-frenado con semiconductores y controles electrónicos (STC, 2021).

En la figura 1.7 se muestra la composición de un tren de nueve carros. Seis de ellos son motrices, es decir, que tienen tracción propia y entre todos arrastran al convoy; ocupan las posiciones 1,3, 4, 6, 7 y 9. Los trenes restantes son remolques (R), es decir sin tracción propia.



Figura 1.7 Representación de un tren de nueve carros.

Fuente: STC (2021).

M: Representa a los carros motrices equipados con cabina de conducción y con tracción propia.

N: Representa los carros motrices con tracción propia y sin cabina de conducción.

R: Representa a los carros remolque.

PR: Representa al carro central que cuenta con el equipo del sistema de piloto automático.



Figura 1.8 Representación de un tren de seis carros.

Fuente: STC (2021).

En la figura 1.8 se muestra la composición de los trenes de 6 carros, 4 de ellos son motrices y 2 remolques. Esta formación puede aumentarse a 9 carros, dependiendo de la demanda de transporte. En la tabla 1.3 se describe la capacidad de cada convoy.

Tabla 1.3 Capacidad de cada tren.

Fuente: STC (2021).

Tren	Sentados	Parados	Total
6 vagones	240	780	1,020
7 vagones	336	1,139	1,475
9 vagones	360	1,170	1,530

El objeto principal de estudio de este trabajo concierne a los trenes de modelo neumáticos, por lo que se hace la descripción del cuerpo de los carros de estos modelos. Se le denomina caja al cuerpo donde viajan los pasajeros. Esta caja va montada sobre dos carretillas portadoras, llamadas boguies, que en el caso de los carros motrices van equipados con dos motores de tracción –un total de cuatro por cada carro motriz-. Los boguies de los carros remolque carecen de motor. Los carros motrices toman la corriente de la barra guía, de 750 volts, mediante las escobillas, situadas entre las dos ruedas de cada boguie (STC, 2021).

Bajo las cajas de los carros motores, entre los boguies, van colocados los equipos que regulan el funcionamiento de los motores de tracción. Bajo la caja de los carros remolque están instalados (STC 2021):

1. El compresor, que produce el aire a presión para el frenado neumático y el cierre de puertas.
2. El moto generador, que genera corriente alterna de 250 volts, destinada al alumbrado de los carros. El compresor y el moto generador trabajan con la corriente de 750 volts, proporcionada por la motriz N adyacente.
3. El bando de baterías, que generan corriente de baja tensión (72 volts), para los circuitos de mando del tren.
4. En el remolque que ocupa la posición intermedia va instalado el captor del piloto automático.

Además, cada boguie consta de dos ejes, en cuyos extremos están fijas dos llantas portadoras. Por dentro de cada una de éstas, existe una rueda de seguridad de acero que, en caso de sufrir una ponchadura, la llanta portadora hace contacto con un riel de seguridad, paralelo a la pista de rodamiento para los neumáticos y sostiene el peso del carro, asegurando su permanencia dentro de las vías. Normalmente esta rueda de seguridad no está en contacto con el riel.

La carretilla cuenta además con cuatro ruedas de guía, más pequeñas, que giran en un plano perpendicular al de las portadoras y van fijas en los brazos de las extremidades del chasis de la carretilla. Estas ruedas se apoyan sobre la barra guía y ayudan a mantener el carro en el centro del carril de rodamiento, amortiguando los vaivenes laterales.

La carretilla tiene un sistema de freno en cada rueda, con zapatas de madera tratada, que hacen presión sobre la banda de las ruedas de seguridad. El freno funciona con aire comprimido. Como ejemplo ilustrativo de la descripción de los trenes, en la figura 1.9 se presenta la vista lateral de un carro.



Figura 1.9 Vista lateral de un carro.
Fuente: STC (2021).

En la tabla 1.4 se describen las características generales del material rodante para los modelos neumáticos.

Tabla 1.4 Características generales del material rodante para los modelos neumáticos.
Fuente: STC (2021).

Carro			
Velocidad comercial:	36 km/h	Peso promedio de carro (N)	
Velocidad máxima:	80 km/h	Peso vacío:	27.8 toneladas
Mantenimiento menor:	Entre 6,000 y 8,000 km	Peso a 4/4 de carga (70 kg/PAS):	39.7 toneladas
Mantenimiento mayor:	Remolques cada 400,00 km	Peso promedio de carro (R)	
Altura riel a techo:	360 m	Peso vacío:	20.8 toneladas
Altura piso a techo:	2.40 m	Peso a 4/4 de carga (70kg/PAS):	32.7 toneladas
Ancho exterior:	2.50 m	Pasajeros sentados (M):	38
Largo motriz con cabina (M):	17.1 m	Pasajeros de pie (M):	132
Largo motriz sin cabina (N):	16.2 m	Pasajeros sentados (N y R):	39
Largo remolque (R):	16.2 m	Pasajeros de pie (N y R):	131
Tipo de frenado:	Reostáticos (eléctrico y/o neumático)		
Peso promedio de carro (M)			
Peso vacío:	28.9 toneladas		
Peso a 4/4 de carga (70kg/PAS):	40.8 toneladas		
Tren de 6 carros (M-R-N-N-PR-M)			
Largo:	99 m	Pasajeros sentados:	232
Peso promedio vacío:	155 toneladas	Pasajeros de pie:	788
Peso promedio a 4/4 de carga (70kg/pas)	226.4 toneladas	Total de pasajeros:	1,020
Tren de 9 carros (M-R-N-N-PR-N-N-R-M)			
Largo:	147.6 m	Pasajeros sentados:	349
Peso promedio vacío:	231.4 toneladas	Pasajeros de pie:	1,181
Peso a 4/4 de carga (70kg/PAS):	338.5 toneladas	Total de pasajeros:	1,530

El STC mantiene una política proactiva de prevención de cualquier emergencia o desastre, ya sean de origen natural como podrían ser geológicos (sismos) o hidrometeorológicos (inundaciones), como de origen antropogénico cuyos orígenes son diversos, variables y extensos (STC, 2017). A pesar de que el metro fue proyectado, diseñado y construido con estos riesgos en mente, observándose en los sistemas hidráulicos como cárcamos de bombeo para evitar inundaciones o sistemas de detección temprana de sismos que permiten tomar acciones de prevención que minimicen daños; mantienen la política de detección oportuna del riesgo con el fin de evitar cualquier emergencia o desastre.

Un factor de riesgo que contribuye a la generación de incidentes es la generación excesiva de basura y falta de limpieza o higiene tanto en las vías como en el interior de la estaciones. Este factor entra en juego gracias a los puestos de comercio que se sitúan en el perímetro de las estaciones, ya que generan basura hacia las rejillas de ventilación, lo cual aunado al riesgo sanitario-ecológico (generación de fauna nociva y transmisión de enfermedades, considerando que parte del público usuario ingiere alimentos en el entorno de la estaciones del STC) se añade el riesgo a incendio por acumulación de basura en las charolas de captación (por debajo de las rejillas) y tirar cigarrillos o cerillos.

El STC describe los tipos de perturbaciones que pueden presentarse durante la operación de la red del STC que interrumpen el servicio. El incidente *cortocircuito por llanta ponchada* es considerado una perturbación del tipo “equipos, instalación y trenes”. El Puesto Central de Control (PCC) es la primera área a la que se notifica, posteriormente las áreas de la STC que están involucradas en la atención del incidente son:

- 1) Mantenimiento de Material Rodante (coordinador), (taller).
- 2) Instalaciones fijas (Coordinación de Control y Operación eléctrica CCOE), (Vías), (Baja tensión).
- 3) Seguridad institucional (BIS), (Centro Estratégico de Operaciones CEO).
- 4) Transportación (Centro Operativo de Estaciones COE), (Gerente), (Subgerente de Control Central), (Subgerente de Líneas)

Las áreas externas al STC que deben de ser informadas en caso de que suceda el incidente son: Ambulancias (en caso de ser necesario), Bomberos (en caso de ser necesario) y RTP. Finalmente las autoridades del STC que deberán tener conocimiento del incidente son: Director General, Subdirectores Generales de Operación y Mantenimiento, Director de Transportación, Director de Material Rodante, Director de Medios y el Gerente de Instalaciones Fijas.

A partir de información recopilada mediante entrevistas al personal del STC durante el 2019, con motivo del desarrollo de propuestas que contribuyan a la reducción significativa de averías en la operación, se determinó qué:

En cuestiones de operación las averías en los trenes son la principal causa de insatisfacción en la calidad del servicio ya que repercuten directamente con retrasos en la circulación de los trenes. Las lluvias, accidentes en las vías principales, accionamiento indebido de palancas de emergencia, entre otras, forman parte de la problemática que afecta la operación y en consecuencia, el servicio del sistema. Para responder a los incidentes relacionados con la operación y servicio de trenes, diversas áreas colaboran entre sí, ya sean áreas internas o externas del STC, así como autoridades del mismo. En el caso del incidente denominado *cortocircuito por llanta ponchada*, intervienen como áreas externas, tanto ambulancias como bomberos.

1.2. EL PROBLEMA POR RESOLVER

Parte del problema surge debido a que el tiempo de atención por parte de áreas externas como son ambulancias y bomberos al incidente *cortocircuito por llanta ponchada* en el STC, depende de la localización de las estaciones de estos servicios y de la ruta que establezcan desde dicha estación hasta la estación de metro más cercana al punto del incidente. A mayor tiempo en la atención al incidente *cortocircuito por llanta ponchada*, corresponde una mayor insatisfacción en la calidad del servicio por parte del público usuario.

1.3. LA SITUACIÓN DESEADA

El área de ingeniería del STC requiere determinar la distancia máxima recorrida de los servicios de emergencia como ambulancias y bomberos desde su localización hasta cualquier punto de la red del metro en donde se presente el incidente *cortocircuito por llanta ponchada*, así como analizar escenarios de optimización. Lo anterior como parte de su sistema de operación y mantenimiento.

1.4. EL OBJETIVO GENERAL

Diseñar, implementar y validar un sistema de simulación-optimización basado en agentes para brindar servicios de emergencia desde hospitales públicos y estaciones de bomberos hacia el lugar donde se presenta el incidente denominado *cortocircuito por llanta ponchada* en la red del metro, mediante la ruta más corta establecida en avenidas principales de la Ciudad de México, utilizando el software NetLogo™.

1.5. LOS OBJETIVOS ESPECÍFICOS

- I. Diseñar e implementar un sistema de simulación-optimización basado en agentes integrado con SIG utilizando el software NetLogo™ para minimizar la distancia a través del modelo de ruta más corta en el metro para el caso del incidente de *cortocircuito por llanta ponchada*, siguiendo la metodología para ABMS.
- II. Analizar los resultados de la simulación-optimización y diseñar escenarios anticipatorios teniendo en cuenta las siguientes variables: probabilidad de cortocircuito, probabilidad de usuarios lesionados, probabilidad de incendio y los recursos con los que cuentan los hospitales y estaciones de bomberos.
- III. Revisar la literatura relacionada con los modelos de optimización y software de simulación enfocada en simulación basada en agentes que se utilizan en la actualidad y que estén relacionados con los temas de estaciones generales y de bomberos además contemplando servicios de emergencia como las ambulancias.
- IV. Elaborar el marco teórico de los tipos de modelos de simulación utilizados en la Investigación de Operaciones.

1.6. METODOLOGÍAS DE LA INVESTIGACIÓN

En este trabajo de investigación, se utilizan principalmente dos metodologías, la primera es aquella que se emplea durante la revisión literaria y la segunda metodología se aplica en el diseño e implementación del sistema basado en simulación-optimización.

La metodología utilizada en la revisión literaria está fundamentada en la metodología planteada en el libro de Machi & McEvoy (2009) donde se distinguen cuatro etapas a seguir para la formulación y planteamiento adecuado de la

revisión literaria. En la figura 1.10, se ilustra en qué consiste la metodología. El argumento de descubrimiento es aquél que discute y explica todo lo que es conocido acerca del tema de interés, mientras que el argumento de abogacía es aquel que analiza y critica el conocimiento adquirido de la síntesis de los datos producidos por el argumento de descubrimiento para responder la pregunta que guía la investigación.

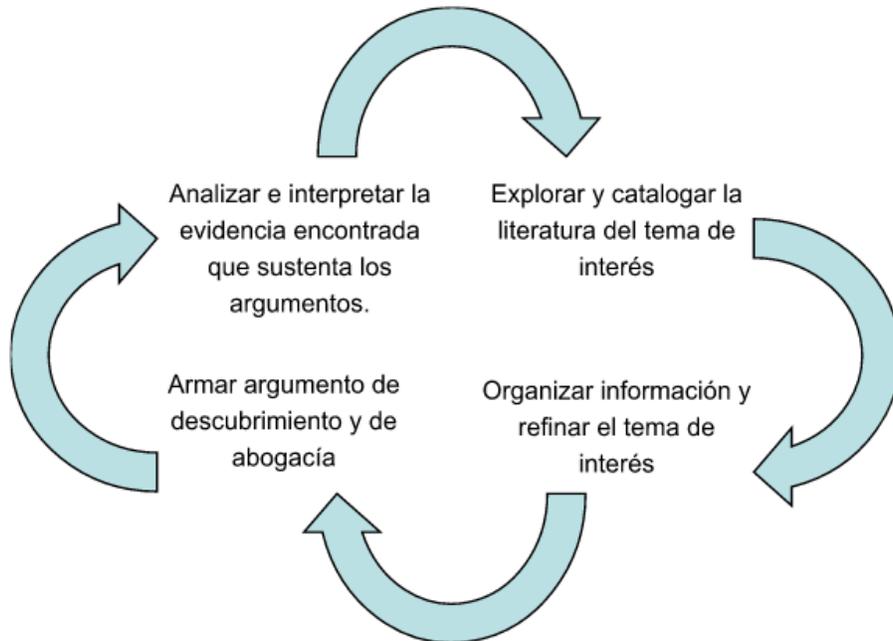


Figura 1.10 Procesos en la revisión de la literatura.
Fuente: Machi & McEvoy (2009).

El diseño e implementación de un sistema de simulación-optimización basado en agentes para brindar servicios de emergencia desde hospitales públicos y estaciones de bomberos hacia el lugar donde se presenta el incidente denominado *cortocircuito por llanta pinchada* en el metro, utilizando el software NetLogoTM, se lleva a cabo con base en la metodología de simulación basada en agentes (Grimm & Railsback, 2005) y para la solución al problema de ruta más corta se utiliza el algoritmo de Dijkstra.

El análisis de los resultados, verificación y validación del modelo se lleva a cabo conforme a la propuesta de (Grimm & Railsback, 2005) en conjunto con (Wilensky & Rand, 2015).

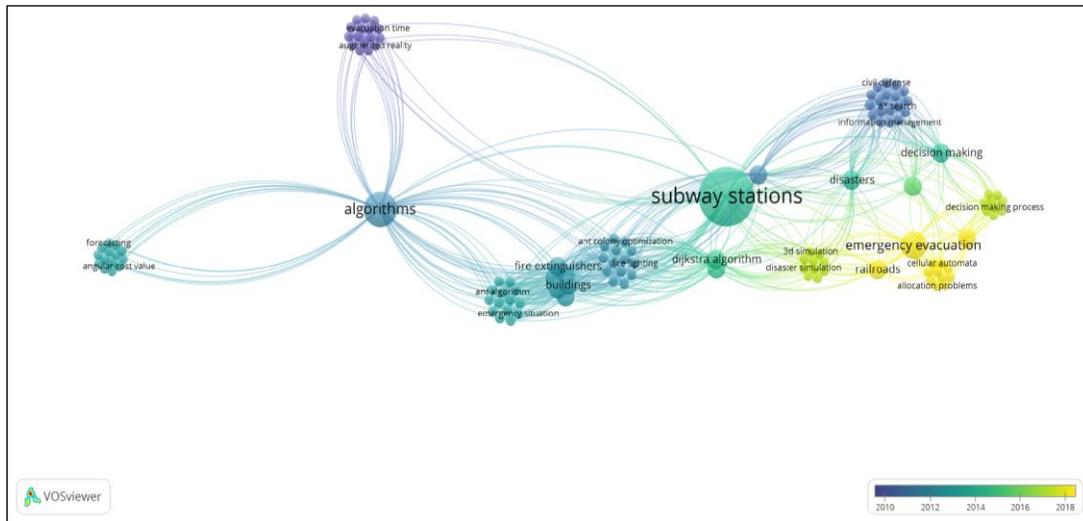


Figura 2.6 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta y estaciones de emergencia, 2012-2018.
Fuente: Elaboración propia (2022).

A partir de la búsqueda “*Subway and Emergency and Stations and Shortest and Path*”, se genera esta red de coocurrencia, figura 2.6, más pequeña que las anteriores debido a que son 10 resultados los que componen esta red.

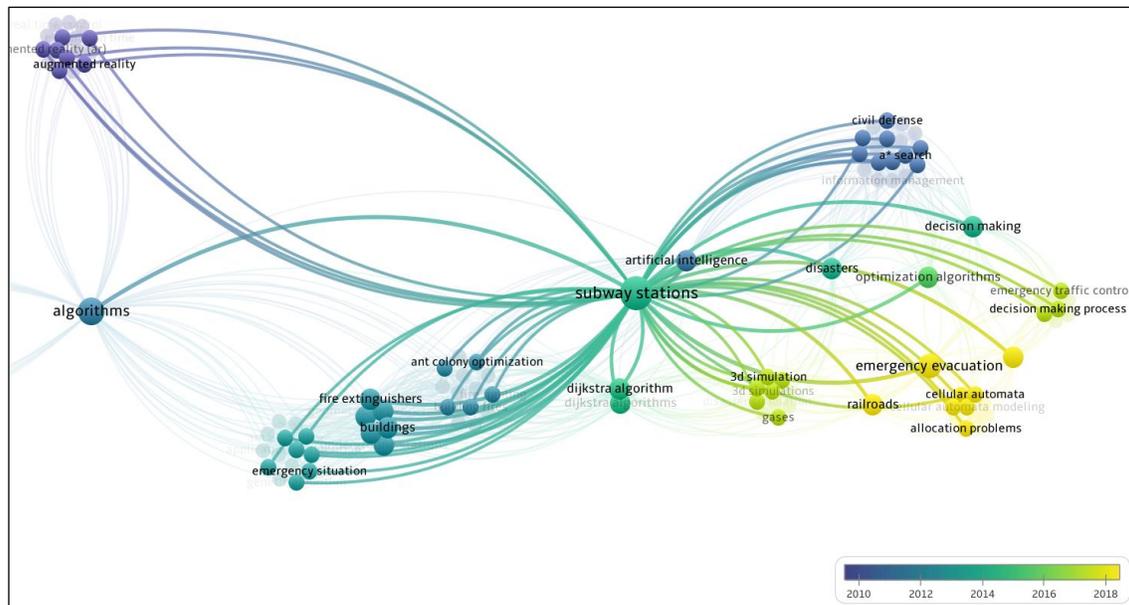


Figura 2.7 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con el problema de ruta más corta, metro y estaciones de emergencia, 2012-2018.
Fuente: Elaboración propia (2022).

Haciendo énfasis en la parte de estaciones de metro, figura 2.7, se puede observar que está relacionado, tomando como lo más antiguo a lo más reciente: (2010) control de flujo, realidad aumentada, tiempos de evacuación, (2012) la inteligencia artificial, (2014) algoritmo de Dijkstra, simulación en 3D, algoritmos de optimización, sistemas de información geográfica, algoritmos genéticos, modelos geométricos de redes, optimización de colonias de hormigas (2018) modelos de autómatas celulares, evacuación de emergencia y problemas de asignación.

- Simulación basada en agentes (*Agent based simulation*): 63,210 resultados.
 - Agent AND based AND simulation AND optimization AND emergency: 116 resultados.
 - Agent AND based AND simulation AND optimization AND subway: 18 resultados.
 - Agent AND Based AND simulation AND optimization AND subway AND emergency: 245 resultados.

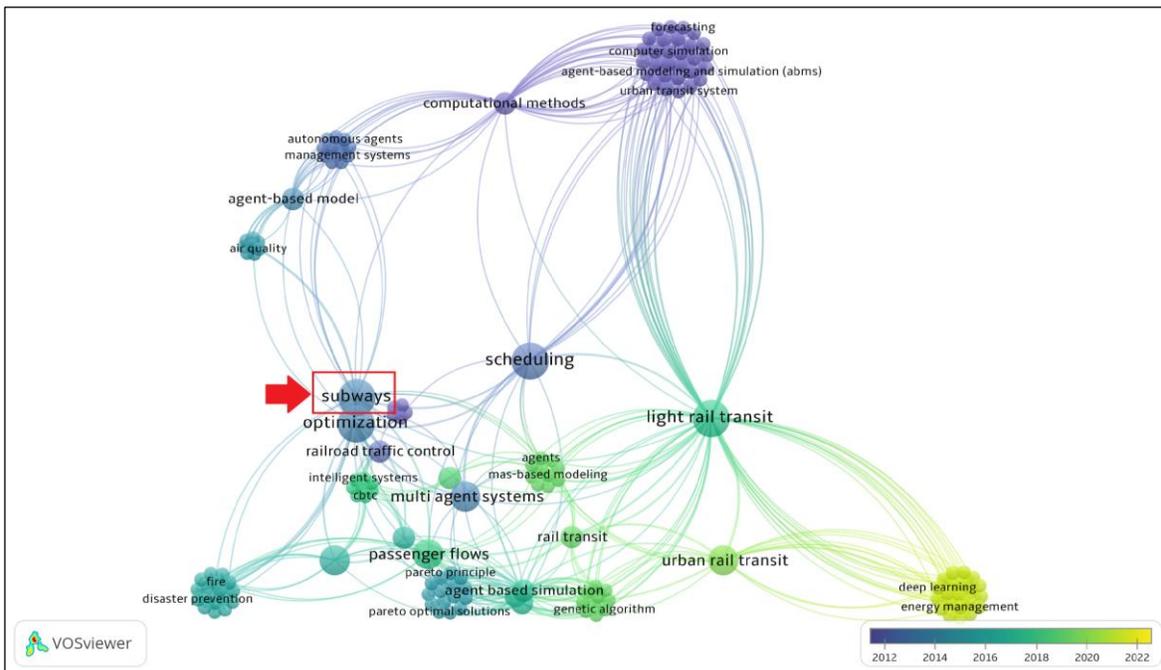


Figura 2.8 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelación y simulación basada en agentes, 2012-2018.

Fuente: Elaboración propia (2022).

Finalmente la red de coocurrencia que se genera a partir de la búsqueda “*Agent and Based and Simulation and Optimization and Subway*”, tiene el fin de relacionar el tema del metro con la simulación basada en agentes, figura 2.8 y figura 2.9.

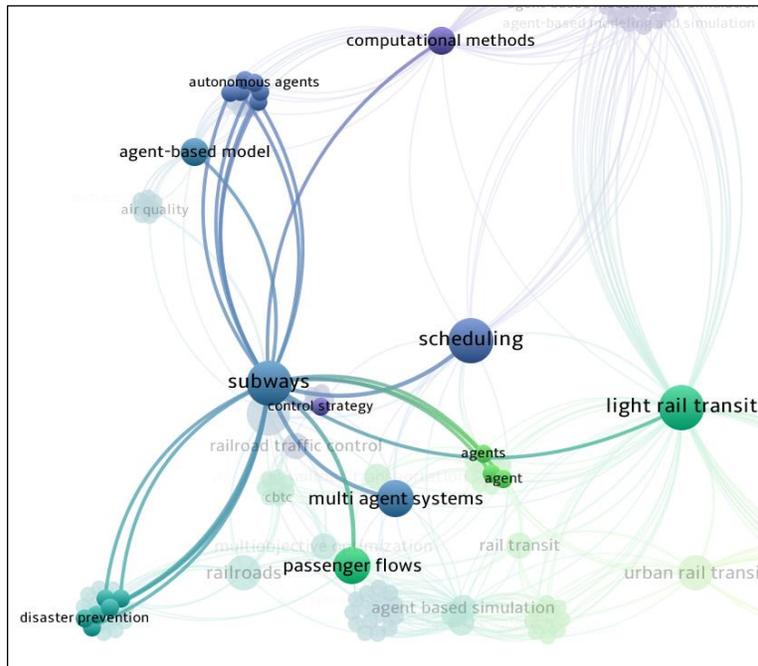


Figura 2.9 Visualización en el software VOSviewer™ de la red de coocurrencias de palabras clave de documentos publicados en relación con modelación y simulación basada en agentes, optimización y metro 2012-2018.
Fuente: Elaboración propia (2022).

Se encuentra que el tema del metro está relacionado con modelos basados en agentes, flujo de pasajeros, sistemas multiagentes, prevención de desastres, agentes autónomos, optimizaciones locales y comunicación entre agentes.

2.2. ANÁLISIS DE LA BÚSQUEDA LITERARIA

Cada una de las líneas de búsqueda literaria relacionó el tema del metro con distintas metodologías como la simulación basada en agentes, la optimización o el modelo de ruta más corta. Se encontraron temas en común entre las distintas búsquedas, lo cual se muestra en la tabla 2.1.

Tabla 2.1 Clasificación de los resultados de la búsqueda de la literatura.
Fuente: elaboración propia (2022).

Tema \ Línea de investigación	Modelos de optimización y estaciones de emergencia	Ruta más corta y estaciones de emergencia	Software de simulación y optimización y modelo de ruta más corta	Metro y estaciones de emergencia	Simulación basada en agentes y optimización en metro
Inteligencia Artificial	Sí	Sí	-	Sí	-
Algoritmos Genéticos	Sí	-	Sí	Sí	Sí
SIG	-	Sí	Sí	Sí	-
Colonia de Hormigas	Sí	-	-	Sí	-
Algoritmo de Dijkstra	-	Sí	-	Sí	-
Evacuación de Emergencia	Sí	Sí	-	Sí	-
Control de flujo	-	-	-	Sí	Sí
Simulación en 3D	-	Sí	-	Sí	Sí
Railroads	Sí	Sí	-	Sí	-
Modelos basados en agentes	-	-	-	Sí	Sí

De la tabla 2.1 se puede apreciar que la búsqueda que relacionó la mayor cantidad de temas fue la del metro y estaciones de emergencia, seguida del modelo de ruta más corta y estaciones de emergencia.

En general se puede reconocer la importancia de los sistemas de información geográfica (SIG o GIS por sus siglas en inglés), el uso de algoritmos genéticos y la inteligencia artificial en relación a los temas de estaciones de metro. En cuestión de modelos de optimización, el de colonia de hormigas fue el modelo que se encontró en al menos dos de las búsquedas relacionadas al tema del metro, sin embargo, también se encontró la programación lineal y problemas de asignación. Se encontró evidencia de que la simulación ha sido utilizada en temas relacionados con estaciones de metro, la simulación en 3D así como la simulación basada en agentes. Además, los temas relacionados a la simulación basada en agentes fueron los algoritmos genéticos y el control de flujo en pasajeros.

El modelo de ruta más corta está relacionado con el tema de estaciones de metro y situaciones de emergencia. El uso de sistemas de información geográfica es un tema relacionado importante. El algoritmo de Dijkstra fue el algoritmo que se relacionó más con el modelo de ruta más corta en la búsqueda realizada. La descripción detallada de este algoritmo se realiza en el capítulo tres de este trabajo de investigación.

2.3. REVISIÓN LITERARIA

En el trabajo de Huang *et al.* (2021) se realiza un análisis de resiliencia en el metro de Chengdu en China. Los autores establecen un modelo dinámico de falla en cascada basado en la teoría de propagación de desastres. Utilizan el algoritmo de

Dijkstra, el cual toma el punto de inicio como el centro y se expande gradualmente hacia afuera capa por capa para encontrar el camino más corto y toma el punto final como el final de la expansión. Además, se determina si existe una estrategia de asignación de recursos óptima para recuperar el mejor rendimiento de recuperación de la red con la menor pérdida de resiliencia.

(Wang, 2021) presenta la teoría de la evacuación de multitudes basada en multiagentes. De acuerdo con este autor, “se espera que China cuente para el 2021 con un kilometraje total de la línea férrea que exceda los 6000 kilómetros”, lo que representa el 15 % del kilometraje total de tránsito ferroviario del mundo. Destaca la importancia del transporte ferroviario urbano como una forma de transporte rápida y cómoda, ya que este alivia la congestión del tráfico en las ciudades. En su revisión literaria (Wang 2021) menciona que “la evacuación segura del tránsito ferroviario urbano es más un método de toma de decisiones en circunstancias discretas” a diferencia de “la evacuación de emergencia, cuya eficiencia es afectada por el comportamiento y estado psicológico de los usuarios. La simulación de evacuación es mediante el software Pathfinder con el modelo de evacuación de peatones el cual está fundamentado en el concepto de fuerza social y considera tres escenarios de un espacio de estación de metro; la plataforma, escaleras y salas de estaciones.

Xie *et al.* (2020) plantean un simulador de evacuación de incendios para investigar el proceso de evacuación de incendios en una plataforma lateral típica de una estación de metro en Guangzhou, China. Este simulador de evacuación de emergencia de incendios, es un modelo de evacuación basado en agentes y un modelo de dinámica de fluidos computacional (CFD), lo desarrollan con el propósito de estudiar la evacuación de incendios considerando la interacción entre el fuego y los humanos. Es un modelo basado en agentes combinado con un modelo de dinámica de fluidos porque como mencionan “El modelado de evacuación de incendios es un campo de la ciencia que se relaciona con la simulación del comportamiento humano durante emergencias de incendios” (Xie *et al.*, 2020) por lo que en un modelo macroscópico como el CFD se pierde el impacto de los factores humanos, del mismo modo si sólo se considera un modelo microscópico como los de autómatas celulares, fuerza social o “lattice gas” se pierde la dinámica del incendio. En China de acuerdo con sus reglamentos y códigos los diseñadores de seguridad contra incendios deben garantizar que todos los pasajeros en el tren y en la plataforma puedan irse oportunamente a regiones seguras en 6 minutos; esto se llama el principio de los "6 minutos", sin embargo, en este estudio proponen utilizar el ASET que es la cantidad de tiempo que transcurre entre el inicio del fuego y el desarrollo de condiciones insostenibles, y el RSET que es la cantidad de tiempo requerida para que los usuarios salgan de un edificio o espacio y lleguen a un lugar seguro. La comparación RSET/ASET permite conocer la influencia de cada parámetro en el desempeño del sistema de seguridad contra incendios a evaluar. Un aspecto importante es que las interacciones entre el fuego y el comportamiento humano se consideran en la simulación: el fuego influye en las condiciones de evacuación al afectar el movimiento y la toma de decisiones de los humanos y bloquear las principales

rutas de salida en casos extremos; por otro lado, los humanos influyen en el fuego abriendo puertas o activando varios dispositivos de protección contra incendios. El modelo de movimiento de agentes es la base del modelo de evacuación. Se utiliza el Modelo de Fuerza Social. El algoritmo de selección de salida del simulador de evacuación de emergencia contra incendios se basa en un modelo de teoría de juegos, los agentes observan las acciones de los demás y seleccionan la salida objetivo a través de la cual estiman que la evacuación será la más rápida. Para crear el modelo de geometría 3D de la estación de metro recogieron datos de campo.

Calabrò *et al.* (2020) proponen un modelo basado en agentes con el objetivo de diseñar rutas de autobuses que cubran la brecha entre la cobertura de transporte público y el número de pasajeros en áreas de baja demanda. El diseño optimizado de rutas de autobuses lo abordan como un problema de enrutamiento de vehículos aplicado al transporte de pasajeros, utilizando optimización de colonia de hormigas (ACO por sus siglas en inglés) para encontrar las rutas de costo mínimo dentro de una red de carreteras. Aplican y verifican su metodología en el caso de estudio de Catania (Italia) donde se está ampliando una línea de metro desde el centro de la ciudad hasta zonas periféricas. Utilizan sistemas de información geográfica o GIS (por sus siglas en inglés) para construir la red de carreteras, seleccionar todas las posibles paradas de autobús y ponderarlas a través de indicadores de accesibilidad, como un indicador de la demanda potencial de transporte. Luego, el algoritmo ACO es desarrollado e implementado en NetLogoTM, con el fin de encontrar un conjunto óptimo de rutas de autobuses, donde la terminal es una estación de metro determinada

En el artículo de la ASCE escrito por Hu *et al.*, (2020) se encuentra un modelo integrado de planeación y operación para implementar una red autobuses que evacuen o transporten a los usuarios de un metro, en caso de interrupción de las operaciones de este. Se trata de una variante del problema de ruteo de vehículos, el cual modelan como un problema de optimización y dada la naturaleza de las restricciones, utilizan simulación y algoritmos genéticos para generar soluciones, además consideran un sistema dinámico en el que las diferentes variables que definen influyen en la simulación para mostrar resultados más precisos. El modelo propuesto y el algoritmo de solución son realizados mediante MATLAB.

Un enfoque diferente se encuentra en el artículo de Wang *et al.*, (2020) donde estudian la evacuación de emergencia en una estación de metro, para este caso analizan la estación Museo Militar la cual es una estación de transferencia para las líneas 1 y 9 del metro de Beijing y su estilo de transferencia es la transferencia de canal. El enfoque que utilizan es el de un problema de asignación de recursos no estándar, donde los pasajeros a evacuar son “demandantes de servicio” y se implementan guías como “instalaciones de servicio”. Debido a que el número de demandantes de servicio es mayor al de las instalaciones de servicio, en este tipo de problema, utilizan el algoritmo de partición de guía basado en el principio de la ruta más corta y considerando la capacidad de tráfico de peatones de salida. Así mismo, utilizan un modelo de autómatas celulares para simular el comportamiento

de evacuación de los usuarios mediante el software Building Exodus y dividiendo la estación del metro en una malla o celdas.

Yin *et al.* (2019) presentan un modelo de optimización, basado en simulación cuyo propósito es dar información a usuarios en una red de tránsito ferroviario. Este modelo de optimización está enfocado a responder cada cuándo, dónde y qué información se le debe proporcionar a los usuarios para que se descongestione la red debido a su alta demanda. La simulación está basada en agentes porque se captura el comportamiento de los usuarios que responden a la información proporcionada. Yin *et al.* (2019) hacen énfasis en la importancia de la optimización combinada con simulación: “En la práctica, el enfoque de optimización basada en simulación (SBO) es una técnica de optimización ampliamente adoptada para abordar problemas de transporte sofisticados.”. El enfoque de simulación basada en agentes que utilizan es el mesoscópico y la función objetivo del modelo de optimización de programación entera, busca minimizar el costo total del viaje del pasajero; por lo que el procedimiento de simulación se utiliza para describir la respuesta del pasajero a la información de guía y las condiciones del tráfico, y luego generar el flujo de tráfico dinámico para la parte de optimización. Para implementar el algoritmo genético y el simulador de flujo de pasajeros utilizan C#, tomando como caso de estudio el metro de Beijing en China. Además, Yin *et al.* (2019) justifican el uso de algoritmos heurísticos en un problema relacionado a un sistema complejo como lo es la red de metro de Beijing:

“La mayoría de los algoritmos de optimización estándar enfrentan dificultades al tratar con redes reales a gran escala. Los algoritmos heurísticos, como los algoritmos genéticos, el algoritmo de colonias de hormigas y la optimización del caos, resultan más adecuados para hacer frente a problemas a gran escala con una demanda computacional intensiva.”

Las operaciones de trenes de metro y cómo influyen en el flujo de pasajeros es el enfoque principal del trabajo desarrollado por Chen *et al.* (2018). En este trabajo mediante el software AnyLogicTM, implementan un modelo multiagente combinado con simulación de dinámica de sistemas para planificar, gestionar y evaluar los flujos de peatones en edificios públicos como estaciones de tren. Para verificar la factibilidad y efectividad del modelo que proponen, el caso de estudio es la estación Hai-Dian-Huang-Zhuang (HDHZ), que es la estación de intercambio de las líneas 4 y 10 en la red de metro de Beijing.

Continuando con los trabajos que abordan el análisis de evacuación de peatones de una estación de metro, está el de Stubenschrott *et al.* (2017) donde destacan la importancia de la simulación para el diseño de infraestructuras a la hora de considerar su rendimiento en servicio, seguridad y egreso de una infraestructura. En este estudio se modela una estación de metro mediante grafos que se generan a partir de datos geométricos, donde elevadores, escaleras eléctricas, escaleras y zonas sobre las que se pueden caminar se representan con enlaces y los puntos de transición son los nodos. Aunado a esto, se utiliza un modelo basado en agentes ya que principalmente los usuarios o peatones se modelan con dos

parámetros; la familiaridad que tienen con la infraestructura que van a evacuar y la preferencia personal a tomar la ruta más rápida o la más conveniente. Consideran un cambio de ruta dinámico al momento en que los agentes cambian su ruta cuando se encuentran con un congestionamiento en el espacio. En el modelo, para encontrar la ruta más rápida, utilizan el algoritmo de Dijkstra.

En el trabajo de Chen *et al.*, (2017) hablan sobre los enfoques (métodos analíticos, simulación y evaluación de riesgos) que se han utilizado para analizar la evacuación de peatones en edificios públicos. Destacan la importancia de evaluar la seguridad de los pasajeros durante el diseño y operación de sistemas de metro. También hacen mención de las metodologías que han sido empleadas para la evacuación de peatones, las cuales incluyen: modelos de autómatas celulares, modelos de dinámica de fluidos, modelos de fuerza social, modelos de “lattice gas” y enfoques experimentales con animales. Así mismo, destacan que los modelos analíticos son apropiados para problemas de evacuación a pequeña escala ya que generalmente es importante para los diseñadores y gerentes de las estaciones de metro conocer si las personas a evacuar son capaces o incapaces de ser evacuadas considerando cierta configuración de la instalación mediante un eficiente y simple método analítico; mientras que se puede utilizar una simulación microscópica para conocer si es seguro o inseguro para las personas evacuar, conocimiento que concierne más a personal técnico e investigadores. Siguiendo esta línea de pensamiento, dividen a la simulación en microscópica y macroscópica, donde la microscópica se encarga de poner atención al comportamiento de los individuos y sus interacciones a diferencia de la macroscópica que considera la situación en general y asume valores agregados. Además, mencionan que las técnicas basadas en múltiples agentes son apropiadas para manejar variables que involucran el espacio, tiempo, intercambio de información, decisiones humanas y el comportamiento humano. Igualmente establecen que un sistema multiagente contiene: entorno, sensores, conocimiento base, reglas de decisión y actuadores. El caso de estudio en el que evalúan y hacen la comparación de cinco distintas metodologías incluyendo la simulación es la estación de metro Xizhimen la cual es una de las grandes estaciones de metro de transferencia en Beijing. Para la simulación utilizan un modelo de fuerza social y AnyLogic™.

El trabajo que realizan Xi *et al.* (2017) se concentra en encontrar las horas de salida óptimas de los usuarios del metro durante las horas pico. A consecuencia de lo anterior, identifican una estrategia para identificar horas de salida relativamente óptimas. En este artículo dividen la simulación de tránsito en macroscópica, microscópica y mesoscópica. La simulación macroscópica estudia las características del flujo de tránsito de manera global. La simulación microscópica se encarga de estudiar las condiciones de las redes de tránsito y los comportamientos detallados de los participantes del tránsito a un nivel de detalle significativo. La simulación mesoscópica propuesta por Prigogine y Andrews, describe los comportamientos de entrada y salida de vehículos en intersecciones y enlaces o en enlaces y paradas pero no incluye los modelos “car-following” y “lane changing”. Hablando en términos de aplicación, tanto la simulación de tránsito

macroscópica como la mesoscópica sirven para la planeación de tránsito y su administración, además la mesoscópica puede presentar sugerencias de rutas. La simulación microscópica en términos de aplicación se encarga de análisis de accidentes y la conducción mediante inteligencia artificial. Continuando con el artículo mencionan que la mayoría de los simuladores de tránsito están basados en modelos multiagentes, ya que estos modelos son una herramienta valiosa para la simulación de sistemas complejos. Mencionan los simuladores de tránsito más comunes como MATSim, (Multi-Agent Transport Simulation) Albatross y ADAPTS (Agent-based Dynamic Activity Planning and Travel Scheduling). Xi *et al.* (2017) mencionan que “La optimización de la ruta de viaje se basa en asignaciones de tráfico estáticas utilizando modelos de equilibrio de usuario (UE) o equilibrio de usuario estocástico (SUE) (Raney *et al.*, 2003)”. Explican que MATSim no es adecuado para analizar “commuting behaviour” ni hay forma de considerar un tiempo de llegada obligatorio, además de que los tiempos de salida son aleatorios. Por estas razones se ven en la necesidad de desarrollar su propia herramienta que nombran CCTSim (Citizen Commuting Transportation Simulation tool) la cual está desarrollada para la optimización de viajes de ciudadanos durante horas pico. El caso de estudio que toman es la ciudad de Shenzhen, ciudad costera con sistema de metro en el sur de China.

Li *et al.* (2016) presentan un trabajo en el que analizan el tiempo de evacuación mediante simulación basada en agentes en la estación de metro Guangzhou en China. El propósito de este trabajo es verificar si la estación cumple con los estándares chinos de diseño del metro en el que el tiempo de evacuación de usuarios al área segura debe ser inferior a 6 minutos y en caso contrario cómo se debe mejorar la estación para una evacuación más rápida. Para realizar esta simulación utilizan el software Pathfinder™. Concluyen con que el diseño de la estación de metro cumple con el “código de requisitos de diseño de metro” ya que el tiempo de salida seguro requerido (RSET) es menor al que marca el código.

Gross (2015) utiliza un modelo de viento a microescala junto con un enfoque de dispersión lagrangiana para simular la propagación de contaminantes atmosféricos liberados accidentalmente en una estación de metro. La contaminación de una multitud de agentes que escapan se especifica en número e intensidad mediante la aplicación de un modelo de simulación basado en agentes. El caso de estudio en el que aplica los modelos en la estación de metro Königsworther Platz en el sistema de metro de Hannover (Alemania).

En el trabajo que desarrollan Lo *et al.* (2014) abordan la importancia de considerar la renovación de estaciones de metro existentes con el fin de mejorar la eficiencia del movimiento de pasajeros en situaciones normales y de emergencia. También remarcan que el movimiento de los peatones generalmente se ha considerado a escala macroscópica en base a la teoría del flujo simple, enfoques de regresión estadística o análisis de flujo de fluidos y cómo si bien dichos métodos pueden mostrar el impacto en el flujo de peatones desde una perspectiva de alto nivel, no pueden incorporar completamente el comportamiento de movimiento de los peatones individuales. Por otro lado, los enfoques microscópicos son la automatización celular, el modelado basado en agentes, etc. "CityFlows" es un

modelo basado en agentes desarrollado por (Lo *et al.*, 2014) para simular el patrón de movimiento en un entorno complejo. “CityFlows” lo combinan con un modelo computacional de dinámica de fluidos “Fire Dynamics Simulator” para modelar la propagación del fuego y el humo en el sistema de metro de Hong Kong.

El objetivo del trabajo desarrollado por Choi *et al.* (2013) fue determinar la relación entre el tiempo de salida y la ruta de salida más corta de un edificio y lo logran mediante la integración de un modelo de comportamiento de peatones y una red geométrica 3D y la evaluación de la estructura del edificio para obtener su tiempo de salida mínimo en caso de emergencia. El proceso de evacuación de un edificio lo simulan utilizando un modelo basado en agentes en el que los agentes son los ocupantes de un edificio y las acciones de estos agentes se basan en un modelo de comportamiento peatonal. La red geométrica 3D proporciona un modelo basado en agentes con información de conectividad de la habitación sobre la distancia desde la ubicación de los ocupantes individuales hasta la ubicación de seguridad. Este trabajo además explica los tipos de modelos de comportamiento del peatón que se utilizan en la modelación basada en agentes: de celdas, de fuerza magnética y de fuerza social, los tres se dividen entre modelos que representan la superficie como una malla (modelo de celdas) o una superficie continua (modelos de fuerza magnética y social). Considerando la eficiencia computacional recomiendan que para simulaciones a nivel micro (evacuación de instalaciones) se consideren de superficie continua y para simulaciones nivel macro (evacuaciones urbanas) se considere la superficie como malla. Además, resaltan la importancia de los sistemas de información geográfica a la hora de proveer una representación continua de la superficie para la modelación basada en agentes.

Un ejemplo de la aplicación de los sistemas de información geográfica o GIS (por sus siglas en inglés) viene del artículo de Wu & Chen (2013) en el que destacan la importancia de los SIG para reducir el tiempo de respuesta a la escena de un incendio. En el estudio se remarcan el uso de dos algoritmos, el primero el de algoritmos genéticos para resolver el problema de p-mediana que se genera al momento de realizar una búsqueda en un piso de una estación de metro para ubicar los equipos de bomberos y el segundo algoritmo es el de colonia de hormigas que resuelve el problema del agente viajero mediante simulación; problema que se plantea cuando los equipos de bomberos necesitan buscar cuarto por cuarto sin repetir y regresando al punto inicial con la ruta más corta. El caso de estudio de este artículo fue una estación de metro de la Administración de Ferrocarriles de Taiwán ubicada en la ciudad de Taipei, Taiwán. El sistema fue establecido en C++ y OpenGL.

El artículo de Kraus *et al.* (2011) habla del proyecto Emergency Management In Large Infrastructures (EMILI) en infraestructuras críticas como aeropuertos y estaciones de metro en donde se utilicen sistemas SCADA (supervisión, control y adquisición de datos) con ayuda de la simulación y entorno de entrenamiento (SITE) al introducir un sistema de apoyo a la toma de decisión. El algoritmo de optimización planteado considera parámetros de evacuación dinámicos y estáticos, además de que simplifica el espacio al considerar pasillos sin paredes y

en caso de edificios con múltiples pasillos largos, la optimización se hace individual y luego se consideran los efectos de uno sobre otro. En este artículo para encontrar la ruta más corta, se utiliza el algoritmo A estrella (A* algorithm) en donde:

“se realiza tratando cada sección como un vértice del grafo. De este modo, el punto medio de la sección se utiliza como ubicación de sección. La distancia entre las secciones se define dentro de la ontología. Como heurística se utiliza la distancia euclidiana (línea aérea) entre el punto medio de un tramo y su salida asignada.” (Kraus et al., 2011, 336).

Para definir la ontología del sistema utilizaron el software abierto Protégé.

El trabajo de Chen & Feng (2009) tiene un enfoque diferente en la evacuación de emergencia ya que no utiliza simulación y en cambio utiliza un algoritmo de control de flujo rápido considerando que la evacuación de edificios se puede considerar como problemas de puertas múltiples estrechas. No utiliza un SIG sino que utiliza representación mediante nodos y arcos. Los algoritmos de control de flujo propuestos fueron probados utilizando el plano de planta del segundo piso de los almacenes Takashimaya en Orchard Road, Singapur, con un programa escrito en C# y utilizando el algoritmo de Dijkstra para encontrar las rutas más cortas.

2.4. CRÍTICA DE LA LITERATURA

Como se mostró en la revisión literaria, existen abundantes trabajos con diferentes enfoques de simulación que han buscado resolver problemas relacionados a sistemas de metro (operativos, instalaciones, trenes) en diferentes países, desde evacuaciones por cuestiones de incendios hasta brindar mayor conectividad al metro. Además, la simulación es una herramienta útil y relevante para modelar sistemas dinámicos relacionados a estaciones de metro como se encontró en los trabajos de Hu *et al.*, (2020), Stubenschrott *et al.* (2017) y Choi *et al.* (2013). Principalmente los dos últimos trabajos destacan la importancia de los modelos basados en agentes.

También se encuentra que se ha utilizado la optimización para resolver problemas de metro como lo hicieron Hu *et al.*, (2020) en el que tratan una variante del problema de ruteo de vehículos para establecer una planeación de autobuses en caso de que hubiera un accidente con un tren o línea de metro y se requiera continuar el servicio o Huang *et al.* (2021) que analizan si existe estrategia de asignación de recursos óptima para recuperar el mejor rendimiento de recuperación de la red con la menor pérdida de resiliencia. Chen & Feng (2009) también utilizan optimización para resolver un problema de control de flujo aplicado a una estación de metro.

Además, autores como Xie *et al.* (2020) y Chen *et al.* (2017) remarcan que los modelos matemáticos de optimización son muy útiles para resolver redes pequeñas y en cambio por complejidad, cuando se trabaja con redes y sistemas

de mayor tamaño, la simulación se convierte en una herramienta sumamente útil y poderosa para entender y analizar estos sistemas complejos.

Se ha utilizado el modelo de ruta más corta junto con la simulación en trabajos relacionados al sistemas de metro como muestran Stubenschrott *et al.* (2017) que consideran un cambio de ruta dinámico al momento en que los agentes cambian su ruta cuando se encuentran con un congestionamiento en el espacio, Huang *et al.* (2021) utilizan el algoritmo de Dijkstra para encontrar el camino más corto cuando falla una estación y se traslada su carga a la estación más cercana, en el trabajo de Lo *et al.* (2014) consideran que los agentes toman la ruta más corta para evacuar, también (Choi *et al.*, 2013) utilizan la ruta más corta para evacuar usuarios en caso de emergencia. Por lo que se demuestra que encontrar la ruta más corta es sin duda un tema de interés crítico relacionado a los sistemas de metro.

Desde el 2015 China ha realizado múltiples trabajos que involucran sistema de metro o transporte urbano ferroviario, esto debido al crecimiento que han tenido diversas ciudades como: Shenzhen, Chengdu o Guangzhou. El sistema de transporte metro, representa una opción viable para mover una cantidad significativa de personas y aliviar el congestionamiento de tránsito como menciona Wang, (2021). Además del crecimiento que han tenido diversas ciudades chinas y la oportunidad de movilidad que representa el sistema de transporte metro, otra razón por la que China ha producido bastante investigación relacionada al transporte urbano ferroviario es la cantidad de kilometraje que posee, (Wang, 2021) menciona que se estimaba que para el 2021 China excediera los 6000 km de kilometraje total de tránsito ferroviario, sin embargo, de acuerdo con (*China Urban Rail Transit Market*, 2021) "...al 31 de diciembre de 2020, 45 ciudades en China operaban un total de 7978 kilómetros (KM) de líneas de tránsito urbano. De estas líneas, 6.300 KM o el 79% eran líneas de metro...", es decir, para el 2020 ya se superaba la cifra de kilometraje total de tránsito ferroviario que se estimaba para el siguiente año, (*China Urban Rail Transit Market*, 2021) explica:

"Este crecimiento fue impulsado por una gran cantidad de ciudades chinas de segundo nivel que llegaron a un punto en el que la Comisión Nacional de Desarrollo y Reforma (NDRC) del gobierno central evalúa la economía local y los ingresos del gobierno pueden respaldar la construcción y operación de un nuevo sistema de transporte ferroviario urbano."

La importancia de los sistemas de información geográfica o GIS (por sus siglas en inglés) en la simulación-optimización se puede apreciar en la mayoría de los artículos que se presentaron en la revisión literaria. Los GIS proveen la oportunidad de establecer en un espacio real los algoritmos y análisis que se plantean como es en el caso de Wu & Chen (2013) que mediante los GIS reducen el tiempo de respuesta a la escena de un incendio o Choi *et al.* (2013) que determinan la relación entre el tiempo de salida y la ruta de salida más corta de un edificio, esto lo logran mediante la integración de un modelo de comportamiento de peatones y una red geométrica 3D, además, realizan la evaluación de la estructura del edificio para obtener su tiempo de salida mínimo en caso de

emergencia. En el trabajo de Calabrò *et al.* (2020) la implementación de los GIS es diferente pero no menos crucial ya que los utilizan para construir la red de carreteras y seleccionar todas las posibles paradas de autobús. Otros artículos si bien no mencionan el uso explícito de los sistema de información geográfica, sí utilizan representaciones espaciales simplificadas como en el caso de Chen & Feng (2009) que simplifican el espacio mediante nodos y arcos.

Se encontraron múltiples trabajos relacionados con la evacuación de estaciones de metro en caso de emergencia. La evacuación de emergencia es un tema fundamental para la seguridad de las personas y cobra vital importancia en instalaciones en donde existe un flujo significativo de personas, como son el caso de estaciones de metro, donde miles de personas hacen uso de las instalaciones diariamente. Los sistemas de metro son un método de transporte eficiente para el traslado de un punto a otro en ciudades cuya alta densidad poblacional ha congestionado vías de automóviles.

El uso del algoritmo de Dijkstra para resolver el modelo de ruta más corta fue presentado en los trabajos desarrollados por Huang *et al.* (2021), Stubenschrott *et al.* (2017) y Chen & Feng, (2009). Cada trabajo tuvo un enfoque y caso de estudio diferente, sin embargo, no hacen mención del porqué el uso de ese algoritmo en lugar de otro, es decir, no se encontró justificación para el uso del algoritmo de Dijkstra sobre el algoritmo A*.

Como se muestra con Calabrò *et al.* (2020) es viable el uso del software NetLogo™ para plantear y resolver mediante simulación basada en agentes, problemas relacionados con sistemas de metro. Previo a este artículo no se encontró evidencia del uso del software NetLogo™ y en cambio se encontró el uso del software Anylogic™ como en el caso de Chen *et al.* (2017) y Chen *et al.* (2018). Otro software que se menciona es el de Pathfinder™ como en el caso de Wang (2021), Li *et al.* (2016) y Xi *et al.* (2017). Sin embargo, es importante remarcar que la aplicación de NetLogo™ fue en cuestión del diseño de redes para comunicar al sistema de metro en estudio con zonas de baja densidad poblacional a diferencia de las aplicaciones de Anylogic™ y Pathfinder™ que fueron utilizados para modelar la evacuación de emergencia de estaciones.

2.5. CONCLUSIÓN DE LA REVISIÓN DE LA LITERATURA

En conclusión, en la revisión literaria se encontró que hay evidencia que apoya el uso de optimización basada en simulación como una técnica de optimización ampliamente adoptada para abordar problemas de transporte sofisticados. Sin embargo, no se encontraron trabajos que aborden el análisis de rutas de servicios de emergencia (ambulancias y bomberos) que responden a incidentes en estaciones de metro. A pesar de esto, se justifica la relevancia del estudio presente con la revisión literaria presentada, donde se destaca que la investigación relacionada a los sistemas de metro ya sea evacuación de estaciones, establecer rutas de autobuses en caso de falla de alguna(s) línea(s)

del metro, diseñar rutas de camiones a zonas de baja densidad poblacional que conecten al sistema de metro o los análisis de resiliencia, son investigaciones relevantes para conocer la seguridad de los usuarios en ciudades cuyas redes de transporte son cada vez más complejas.

MARCO TEÓRICO

2.6. TIPOS DE MODELOS DE SIMULACIÓN EN INVESTIGACIÓN DE OPERACIONES

El término simulación significa diferentes cosas en diferentes disciplinas y abarca una amplia variedad de enfoques, incluida la simulación de Monte Carlo, la micro simulación y el juego de roles o simulaciones de "humano en el circuito". De acuerdo con (Brailsford *et al.* 2019) "...entre todos los enfoques de simulación, la simulación de eventos discretos (DES), la dinámica de sistemas (SD) y la simulación basada en agentes (ABS) fueron los más utilizados en la investigación de operaciones (OR) para modelar problemas de negocios."

Brailsford et al. (2019) presentan una metodología para el modelado de simulación híbrida que bien puede considerar para cada tipo de simulación:

"...definición del problema, selección de métodos, modelado conceptual, implementación informática, recopilación de datos, parametrización, verificación y validación, desarrollo de escenarios a probar, experimentación, análisis y presentación de resultados e (idealmente) uso de los resultados para informar una decisión del mundo real, lo que permite, por supuesto, la iteración hacia atrás entre etapas."

En otras palabras, la metodología que (Brailsford *et al.* 2019) proponen para el modelado de simulación híbrida está conformada por cuatro etapas:

- 1) Problema del mundo real
 - a) Área de aplicación.
 - b) Tipo de estudio (estudio de caso, marco + aplicación, teórico).
- 2) Modelado conceptual
 - a) Modelo conceptual informado, incluida la descripción de los subsistemas (con una indicación del enfoque de simulación apropiado), y las interrelaciones entre los subsistemas (tipo de información intercambiada y frecuencia)
 - b) Tipo de modelo (ABS+DES, ABS+SD, DES+SD, ABS+DES+SD).
 - c) Tipo de hibridación (enriquecimiento, secuencial, interacción, integración).
- 3) Modelado por computadora
 - a) Proceso de integración del modelo (manual, utilizando herramientas intermedias, automatizado).
 - b) Datos de entrada (mundo real, ilustrativo, mixto).
 - c) ¿Verificación reportada?

- 4) Solución y entendimiento
 - a) Validación (estadística, validez aparente, otra).
 - b) Nivel de implementación (prueba de concepto, potencial, real).

2.7. *SIMULACIÓN DINÁMICA DE SISTEMAS*

De acuerdo con (Dangerfield, 2014) “El campo de la dinámica de sistemas (SD por sus siglas en inglés) se conoció inicialmente como dinámica industrial, que reflejó sus orígenes en la simulación de problemas de la cadena de suministro industrial.” Así mismo, Dangerfield (2014) menciona las características de la Simulación Dinámica de Sistemas (SD):

- Abordan problemas considerando agregados (de productos, personas, etc.) y no entidades individuales (DES) o como agentes individuales (ABS).
- Principalmente reflejan la dinámica de un sistema como si tuviera causas endógenas (internas): el cambio con el tiempo proviene de los límites del sistema debido a la información efectos de retroalimentación e interacciones de componentes, aunque el estímulo inicial para esas dinámicas pueden ser exógenas (externas).
- Distinguen cuidadosamente entre los flujos de recursos y los flujos de información que hacen que esos flujos de recursos aumenten o disminuyan. Esta es una característica fundamental (y poderosa) de la metodología, lo que significa que dichos modelos pueden usarse para diseñar y evaluar sistemas de información, así como el enfoque más habitual en los sistemas de recursos.
- Se supone que los flujos son continuos y se rigen por lo que son, en efecto, ecuaciones diferenciales ordinarias. Los modelos de dinámica de sistemas (SD) pertenecen a la categoría más amplia de modelos de simulación continua.
- Aunque se incluyen las tasas de flujo, los modelos SD se ocupan principalmente del comportamiento de las existencias o acumulaciones en el sistema. Estos se describen mediante ecuaciones integrales. “Forrester ha dicho que la diferenciación no existe en la naturaleza, solo la integración.” (Dangerfield, 2014). Los modelos matemáticos caracterizados por ecuaciones diferenciales deben resolverse para determinar los valores de las existencias; la dinámica del sistema pone las variables de existencias en primer plano.
- No ignoran las variables suaves (como la moral o la reputación) cuando se sabe que tienen una influencia causal en el sistema.

Posteriormente (Dangerfield, 2014) afirma que: “La retroalimentación (feedback, en inglés) es un bloque de construcción esencial de SD mediante el cual la información sobre el estado actual del sistema se utiliza para regular los controles sobre los flujos de recursos y subraya el punto de vista endógeno.”

2.8. SIMULACIÓN DE EVENTOS DISCRETOS

De acuerdo con (Brailsford *et al.*, 2014) “La simulación de eventos discretos (DES, por sus siglas en inglés) surgió en gran medida del deseo de modelar sistemas de fabricación. Basados en los métodos de Monte Carlo, se desarrolló la DES para mejorar el diseño y la operación de las plantas manufactureras.” DES es visto como un enfoque de simulación convencional, en el campo de la investigación de operaciones (IO), cuyas aplicaciones incluyen: salud, industrias de servicios, transporte, almacenamiento, cadenas de suministro, defensa, sistemas informáticos y gestión de procesos comerciales.

Brailsford *et al.* (2014) continúan explicando que “...la DES modela los sistemas de colas a medida que avanzan en el tiempo. Al hacerlo, representa el mundo como entidades que fluyen a través de una red de colas y actividades. Donde los recursos se comparten entre actividades”. Las características de un modelo DES son las siguientes (Brailsford *et al.*, 2014):

- **Entidades:** elementos individuales que fluyen a través del sistema, por ejemplo, dispositivos en una planta de fabricación, personas en un sistema de servicios u hospital, paquetes de información en una red informática, vehículos en un sistema de transporte, pedidos en una cadena de suministro.
- **Colas o líneas de espera:** áreas en las que las entidades esperan a que se les trabaje, por ejemplo, zonas de influencia y almacenes, inventario, áreas de espera, listas de espera, colas de llamadas telefónicas
- **Actividades:** realizan trabajo en entidades, por ejemplo, máquinas, viajar, moverse, servir, imprimir.
- **Recursos:** necesarios para estar presentes para operar actividades, por ejemplo, operadores, equipos, médicos.

Además Brailsford *et al.* (2014) mencionan que otros dos elementos clave de los modelos DES son: el mecanismo de manejo del tiempo y los métodos para el muestreo aleatorio.

2.9. MODELACIÓN BASADA EN AGENTES

Para hablar sobre la modelación basada en agentes primero se tiene que conocer sobre los sistemas complejos. De acuerdo con Wilensky & Rand (2015) la teoría de los sistemas complejos desarrolla principios y herramientas para dar sentido a la complejidad del mundo, así como para observar distintos fenómenos y define los sistemas complejos como sistemas que están compuestos por múltiples elementos individuales que interactúan entre sí pero cuyas propiedades o comportamiento agregado no es predecible a partir de los elementos mismos. El campo de los sistemas complejos o la ciencia de la complejidad surgió a mediados de la década de 1980 y nació de una variedad de campos dispares, desde la economía hasta la física y la ecología. En muchos sistemas complejos, los

procesos probabilísticos y aleatorios contribuyen e incluso son esenciales para la creación de orden.

Los sistemas complejos se caracterizan por fenómenos emergentes (patrones que parecen bastante complejos a menudo pueden generarse mediante reglas simples). “La emergencia es una propiedad exhibida clásicamente por muchos modelos basados en agentes, y ocurre cuando un atributo que puede describirse a nivel de sistema no está específicamente codificado a nivel individual.” (Wilensky & Rand, 2015). Sin embargo, la emergencia, presenta dos desafíos fundamentales y distintos. La primera dificultad radica en tratar de averiguar el patrón agregado cuando se sabe cómo se comportan los elementos individuales (comprensión integradora). Una segunda dificultad surge cuando se conoce el patrón agregado y se trata de encontrar el comportamiento de los elementos que podrían generar el patrón (comprensión diferencial o comprensión composicional).

Una metodología importante de la ciencia de la complejidad es el modelado basado en agentes. La modelación basada en agentes consiste en tres elementos clave. Los agentes, un entorno y las interacciones que tienen los agentes entre sí y con su entorno. “Un agente es una entidad computacional que posee propiedades o variables y valores” (Wilensky & Rand, 2015). Los agentes además poseen reglas de comportamiento y pueden interactuar entre sí o con su entorno, el entorno es el paisaje en el que interactúan los agentes y puede ser geométrico, basado en redes o extraído de datos reales.

La complejidad de las interacciones agente-agente y agente-entorno es variable, ya que pueden cambiar con el tiempo. Un factor que contribuye a la complejidad de las interacciones, además del tiempo, es la disponibilidad de información con la que cuentan los agentes. Las acciones de los agentes pueden cambiar dependiendo de la información que adquieren con el tiempo. Como resultado de estas interacciones, los agentes pueden actualizar su estado interno o realizar acciones adicionales.

De acuerdo con Wilensky & Rand (2015) los modelos basados en agentes tienen ocho usos principales:

- I. **Descripción:** Un modelo es descriptivo de un sistema del mundo real, una simplificación que no contiene todos los detalles e inconsistencias.
- II. **Explicación:** Los modelos son explicativos en el sentido de que señalan los mecanismos esenciales que subyacen a un fenómeno.
- III. **Experimentación:** Los modelos son explicativos en el sentido de que señalan los mecanismos esenciales que subyacen a un fenómeno.
- IV. **Fuentes de analogías:** Debido a que son simplificaciones de la realidad, permiten que se encuentren similitudes con otras simplificaciones aún si los fenómenos modelados aparentemente son ampliamente diferentes.
- V. **Comunicación/Educación:** Los modelos brindan una herramienta educativa que encapsula el conocimiento que puede no estar fácilmente disponible a partir de la observación del mundo real.

- VI. **Proporcionar objetos focales o piezas centrales para el diálogo científico:** Los modelos basados en agentes permiten discutir el comportamiento de los agentes ya que se especifican los mecanismos importantes que generan el comportamiento del agente. En términos más generales, la “caja de vidrio” de los modelos basados en agentes brinda una representación clara del problema que se examina y, por lo tanto, es útil no solo dentro de la comunidad científica sino también en otros ámbitos, como el análisis de políticas.
- VII. **Experimentos mentales:** Los modelos a veces pueden presentar nuevos fenómenos que no necesariamente se relacionan con algún fenómeno del mundo real, sino que son experimentos mentales sobre posibles cálculos. Algunos ejemplos clásicos de este tipo de modelos son los autómatas celulares, los fractales y los enjambres de partículas.
- VIII. **Predicción:** A menudo se utilizan modelos para pensar en posibles escenarios futuros, los modelos basados en agentes se pueden usar de esta manera. Sin embargo, como cualquier herramienta de modelado, las predicciones de un modelo basado en agentes dependen de la precisión de sus datos de entrada. Esto es especialmente cierto cuando los eventos de interés son los resultados de sistemas complejos, para los cuales pequeños cambios en las entradas a menudo pueden conducir a resultados muy diferentes.

Como establece Wilensky (2015), una característica de la modelación basada en agentes (ABM) es que permite modelar poblaciones heterogéneas al modelar individuos, a diferencia de la modelación basada en ecuaciones (EBM) que frecuentemente establece suposiciones sobre homogeneidad. Aunado a esto, al modelar individuos, las interacciones que se generan son del tipo discretas en lugar de continuas.

Otra perspectiva de contraste entre la modelación basada en agentes (ABM) y la modelación basada en ecuaciones (EBM) consiste en que con la ABM no se requiere conocer los fenómenos agregados, es decir, se puede construir el modelo a partir de simples reglas para entidades simples y observar el resultado agregado. La EBM requiere de conocimiento de ecuaciones diferenciales y conocer previamente el fenómeno agregado, descrito mediante ecuaciones, para poder probar la hipótesis contra el resultado agregado. “Por lo tanto, incluso si no tiene hipótesis sobre cómo interactuarán las variables agregadas, aún puede construir un modelo y generar resultados” (Wilensky, 2015).

Además de la característica de la ABM de poder generar modelos a partir de reglas simples de interacción, agente-agente y agente-entorno que permiten observar el comportamiento agregado de un sistema lo que resulta en una descripción más parecida a la realidad a diferencia de una descripción basada en agregados, es que resulta más fácil de describir el modelo y su funcionamiento a quien no está familiarizado con una modelación matemática avanzada. Esto representa una ventaja porque el modelo puede ser entendido por todas las partes interesadas durante el proceso de modelado, lo que a su vez ayuda a mejorar la verificabilidad del modelo. “Este enfoque de modelado de “caja de cristal” (Tisue & Wilensky, 2004) permite a todas las partes interesadas hablar sobre el modelo hasta sus componentes más básicos.” (Wilensky, 2015).

2.10. METODOLOGÍA PARA EL DESARROLLO DEL MODELO DE SIMULACIÓN BASADO EN AGENTES.

La metodología principal utilizada para el desarrollo del modelo de simulación basada en agentes proviene de Wilensky & Rand (2015) y como metodología complementaria se utiliza la planteada por Brailsford *et al.* (2019). Ambas metodologías comparten elementos clave para la modelado de simulación basada en agentes.

La figura 2.11 describe la metodología utilizada (Wilensky & Rand, 2015, 1-45). En esta metodología se establece que primero se defina el modelo conceptual utilizando el lenguaje común para describir las propiedades de los agentes, su entorno y las interacciones que tienen. Posteriormente, una forma de hacer que el modelo conceptual más generalizable y obtener información sobre el comportamiento del sistema complejo que se está analizando, sería implementar el modelo en forma computacional.

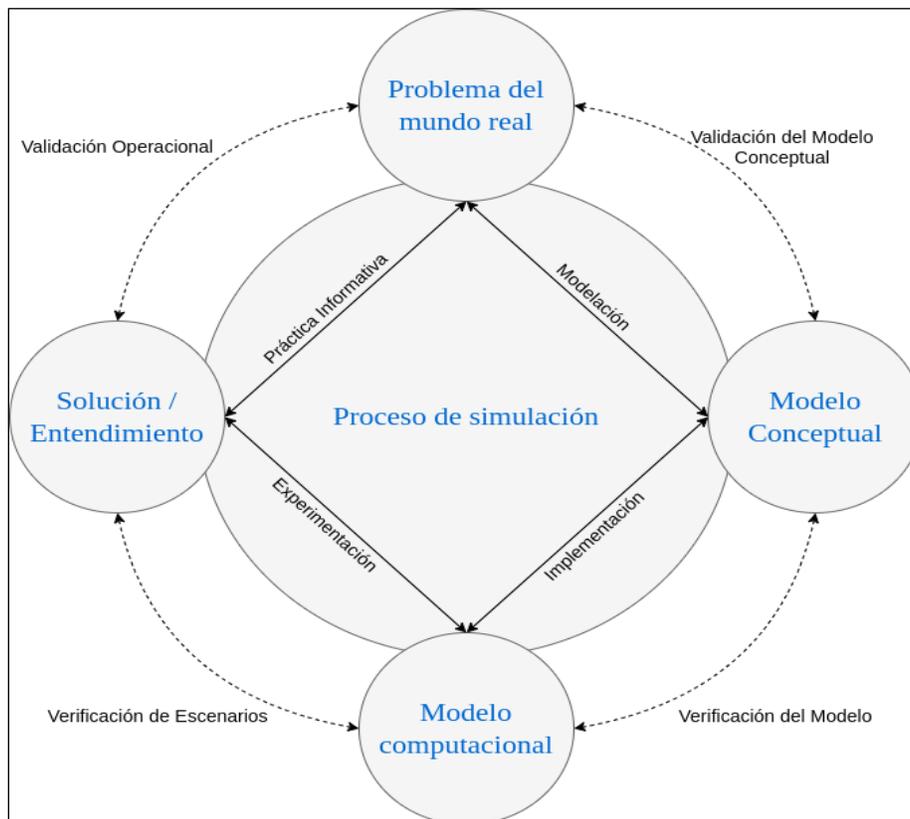


Figura 2.10 Metodología para el desarrollo del modelo de simulación basada en agentes.

Fuente: Adaptado de Wilensky & Rand (2015).

“Un modelo computacional es un modelo que toma ciertos valores de entrada, manipula esas entradas de forma algorítmica y genera salidas.” (Wilensky & Rand, 2015, 45). La forma en cómo se transforma el modelo conceptual que describe los procesos, comportamiento y entorno de un sistema, a un modelo computacional es mediante la descripción algorítmica del modelo conceptual.

“La verificación es el proceso de comparar un modelo implementado con su modelo conceptual asociado e investigar si el modelo implementado es fiel al modelo conceptual.” (Wilensky & Rand, 2015, xix). El objetivo de la verificación es la eliminación de "errores" del código, ya que a medida que un modelo basado en agentes crece, se vuelve más difícil simplemente mirar su código para determinar si realmente está realizando la función con la que fue diseñado. (Wilensky & Rand, 2015, 312) mencionan que una pauta general para habilitar la verificación del modelo implica construir el modelo de manera sencilla o simple para comenzar, expandiendo los niveles de complejidad del modelo solo según sea necesario. Si un modelo es simple o sencillo inicialmente, es más fácil de verificar que un modelo complejo. Del mismo modo, si las partes complementarias añadidas al modelo también son de naturaleza incremental (construyendo hacia su pregunta de interés en lugar de tratar de desarrollar toda la elaboración del modelo a la vez), esos complementos también serán más fáciles de verificar (y por extensión, el modelo del que forman parte). Aun así advierten que se debe tener en cuenta que incluso si se verifican todos los componentes de un modelo, es posible que el modelo en sí no lo sea, ya que pueden surgir complicaciones adicionales de las interacciones entre los componentes del modelo.

“La validación es una comparación de un modelo implementado con el mundo real, para ver si los resultados del modelo implementado nos dan una idea del fenómeno del mundo real correspondiente.” (Wilensky & Rand, 2015, xix). La validación, por su naturaleza, es compleja, multinivel y relativa. “Los modelos son simplificaciones de la realidad; es imposible que un modelo exhiba todas las mismas características y patrones que existen en la realidad”. (Wilensky & Rand, 2015, 326). La validación de un modelo comienza en el nivel más bajo del modelo, en otras palabras, la validación comienza comparando las acciones y los datos de los agentes en un modelo con los que existen en el mundo real (mismas propiedades, movimiento, rumbo, ubicación) y continúa a través de las capas de complejidad del modelo. La importancia del proceso de validación radica en el uso que se le da al modelo, ya que si se utiliza un modelo para hacer afirmaciones sobre el mundo, tiene que haber una conexión defendible entre el modelo y la realidad que se está simplificando. Durante el proceso de validación, es importante tener en cuenta las preguntas del modelo conceptual y validar los aspectos del modelo que se relacionan con estas preguntas.

“La replicación es la reproducción del resultado de un modelo publicado por un científico o desarrollador de modelos por otro científico o desarrollador de modelos.” (Wilensky & Rand, 2015, xx). La replicación representa entonces un componente fundamental del método científico ya que para que un experimento sea aceptable para la comunidad científica se deben compartir los detalles de cómo fue realizado el experimento. La replicación es entonces un proceso entre el

modelador y la comunidad científica ya que equipos posteriores de científicos deben realizar experimentos ellos mismos para determinar si sus resultados son lo suficientemente similares para confirmar los resultados originales con el fin de confirmar el hecho de que el experimento no dependía de ninguna condición local. De acuerdo con (Wilensky & Rand, 2015, 336) “La replicación es una parte importante del proceso científico y es tan importante dentro del ámbito de los modelos computacionales como lo es dentro del ámbito de los experimentos físicos”. La importancia de replicar un modelo computacional radica en que aumenta la confianza en la verificación del modelo, ya que una nueva implementación del modelo conceptual ha arrojado los mismos resultados que el original. La replicación también puede ayudar en la validación del modelo, ya que requiere que los replicadores del modelo intenten comprender las elecciones de modelado que se tomaron y cómo los modeladores originales vieron la coincidencia entre el modelo conceptual y el mundo real.

2.11. ALGORITMO DIJKSTRA

El científico informático Edsger Dijkstra desarrolló un algoritmo para encontrar el camino más corto para cualquier gráfico dirigido ponderado con pesos no negativos. En (Flores de la Mota, 2010) se presenta el algoritmo Dijkstra para obtener la arborescencia de las rutas más cortas de raíz s en una red dirigida con costos no negativos en los arcos, donde:

“Sea $G = [N, A]$, una red dirigida y s un nodo en N ; entonces a s se le llama raíz de G , si existe una trayectoria de s a i para toda i en N . Una arborescencia de G es un árbol expandido de G que contiene un nodo que es raíz. Una arborescencia de rutas más cortas de G es aquella arborescencia tal que la única ruta de s a i , para toda i elemento de N , es una ruta más corta de s a i .”

De acuerdo con (Flores de la Mota, 2010), el propósito del algoritmo Dijkstra es obtener la arborescencia de las rutas más cortas de raíz s en una red $G = [N, A, d]$ con costos no negativos en los arcos; este método se basa en la asignación de etiquetas “permanentes” a los nodos para los cuales ya se conocen las longitudes de las rutas más cortas de la raíz a ellos. Sea este conjunto de nodos. Las etiquetas de los nodos representan precisamente las longitudes de las rutas más cortas buscadas. Los nodos restantes se etiquetan “temporalmente” en una cota superior de la longitud más corta de la raíz al nodo etiquetado.

Descripción del algoritmo

- Paso 1: (Iniciación de etiquetas). Sea $d(s) = 0$ y márchese esta etiqueta como permanente. Sea $d(x) = \infty$, para todo $x \neq s$ y considérense estas etiquetas como temporales. Sean $a(x) = x$ (estas etiquetas indicarán el predecesor de en la arborescencia). Sea $p = s$.
- Paso 2: (Actualización de etiquetas). Para todo $x \in \Gamma^+(p)$ todo que tenga etiqueta temporal, actualizar la etiqueta de acuerdo con:

$$d(x) = \min\{d(x), d(p) + d(p, x)\}$$

- Si $d(x)$ se modificó, hacer $a(x) = p$. Sea x^* tal que $d(x^*) = \min\{d(x) | d(x) \text{ es temporal}\}$. Si $d(x^*) = \infty$ terminar. En este caso no existe arborescencia alguna de raíz s . En otro caso, marcar la etiqueta $d(x^*)$ como permanente. Sea $p = x^*$.
- Paso 3:
 - (Si sólo se desea la ruta de s a t). Si $p = t$, terminar: $d(p)$ es la longitud del camino más corto. Si $p \neq t$, ir al paso 2.
 - (Si se desea la arborescencia). Si todos los nodos tienen etiquetas permanentes, terminar; ésta es la longitud deseada del camino y el conjunto de arcos $\{a(x), x\}$ forman la arborescencia de caminos más cortos. En otro caso, ir al paso 2.

En Huang et al. (2009), el pseudocódigo del algoritmo de Dijkstra se presenta a continuación:

Dijkstra (Graph G , Vertex S)

1. Initialize (G, s);
2. Priority_Queue $minQ = \{\text{all vertices in } V\}$;
3. while ($minQ \neq \emptyset$) do
4. Vertex $u = \text{ExtractMin}(minQ)$; // minimum est(u)
5. for (each $v \in minQ$ such that $(u, v) \in E$)
6. Relax (u, v);
7. end for
8. end while

2.12. SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Los sistemas de información geográfica (Geographical Information Systems, GIS) son herramientas computacionales basadas en el manejo de capas de temáticas cuya característica principal es la referencia a un sistema de coordenadas geográficas. En relación con el acceso a los usuarios, existen GIS de acceso restringido y acceso abierto.

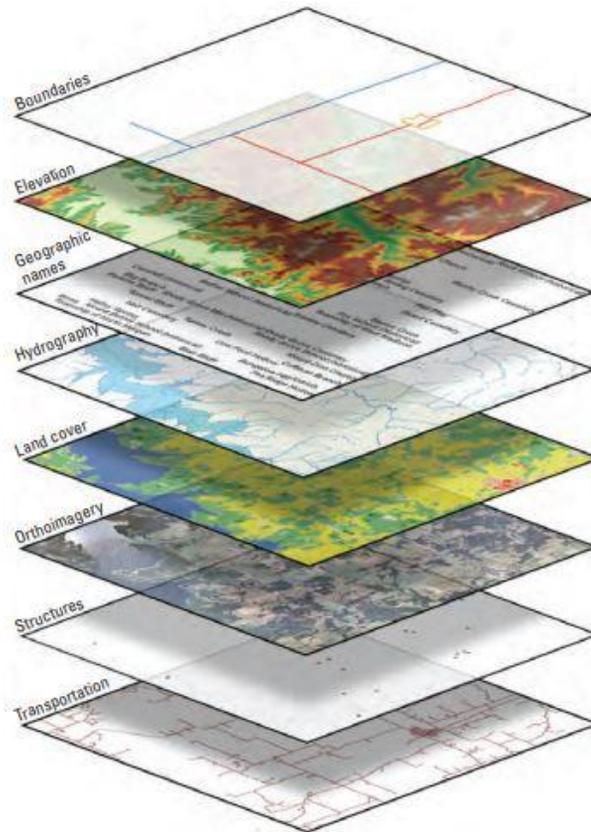


Figure 1. Eight base layers of *The National Map*.

**Figura 2.11 Ejemplo de capas temáticas sobrepuestas en QGIS.
Fuente: USGS (fecha acceso 2022).**

La integración de GIS con modelos de simulación basados en agentes permite a los desarrolladores de modelos, pensar en la forma en que los agentes interactúan y cambian en el espacio y tiempo (Crooks *et al.*, 2019). El espacio en los modelos basados en agentes contiene a los agentes propiamente y define las interrelaciones espaciales entre los mismos y el control de sus movimientos.

De acuerdo con Kent State Online, (2021) y GISGeography (20 de junio de 2022) los componentes de un Sistema de Información Geográfica establecen un marco fundamental sobre el cual todos los tipos de información geográficamente referenciada pueden ensamblarse lógicamente y explorarse científicamente, siendo los siguientes:

1. Recursos humanos (personas): aquellos que administran el sistema y desarrollan planes para aplicarlo a problemas del mundo real.
2. Métodos: un GIS operará acorde con un plan bien diseñado y reglas claras.
3. Datos: los datos geográficos y los datos tabulares relacionados pueden recopilarse internamente o comprarse a un proveedor de datos comercial.

4. Programa (Software): el programa que proporciona las funciones y herramientas necesarias para almacenar, analizar y mostrar información geográfica.
5. Equipo (Hardware): la computadora o equipo con en el que se ejecuta el GIS.

2.13. QGIS

En este trabajo, se utiliza QGIS un sistema de información geográfica de código abierto que de acuerdo con (*Descubre QGIS*, septiembre 2022):

“QGIS es un Sistema de Información Geográfica (SIG) de Código Abierto licenciado bajo GNU - General Public License . QGIS es un proyecto oficial de Open Source Geospatial Foundation (OSGeo). Corre sobre Linux, Unix, Mac OSX, Windows y Android y soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos.”

QGIS proporciona una creciente gama de capacidades a través de sus funciones básicas y complementos. Puede visualizar, gestionar, editar y analizar datos, y diseñar mapas imprimibles.

La versión con la que se trabajó para desarrollar este trabajo de investigación es: 3.24.1-Tisler.

CAPÍTULO III: UN SISTEMA BASADO EN SIMULACIÓN-OPTIMIZACIÓN

3. UN SISTEMA BASADO EN SIMULACIÓN - OPTIMIZACIÓN

3.1. JUSTIFICACIÓN DE LA MODELACIÓN BASADA EN AGENTES (ABSM)

Para este caso de estudio, dado que se sugiere diseñar e implementar un sistema de simulación-optimización basado en agentes para brindar servicios de emergencia desde hospitales públicos y estaciones de bomberos hacia el lugar donde se presenta el incidente denominado *cortocircuito por llanta ponchada* en el metro, los trenes serán modelados como agentes dado que deberán ser capaces de controlar su trayectoria, paradas, etc., con base en lo que perciban de su entorno operativo como aproximación de otros trenes, usuarios, etc.

De acuerdo con Wilensky & Rand (2015) el modelado basado en agentes es poderoso porque su ontología básica es paralela a la ontología del mundo real. Esto permite que mediante el uso de la modelación basada en agentes se pueda describir y definir sistemas complejos de manera natural, ya que no es necesario definir las propiedades globales del sistema y en cambio se pueden observar a través de la simulación. Además Wilensky & Rand (2015) mencionan que los modelos basados en agentes son más fáciles de comprender a comparación de los modelos matemáticos que estudian el mismo fenómeno debido a que los modelos basados en agentes son construidos a partir de objetos individuales y reglas simples para su movimiento o comportamiento, a diferencia de modelos de ecuaciones que están construidos a partir de símbolos matemáticos, en otras palabras, es más fácil y cercano al discurso natural humano para describir una experiencia en términos de interacciones de individuos a diferencia de razones de cambio de agregados como sucede en ecuaciones diferenciales.

Aunado a esto, se considera la movilidad de la CDMX como un sistema complejo, ya que está compuesta por diversos subsistemas como el STC, automóviles privados, medios de transporte públicos como camiones, etc. Debido a su complejidad, no se conoce el comportamiento agregado de todos estos sistemas, lo cual es un requisito para utilizar métodos de modelación tradicionales como ecuaciones diferenciales (modelos basados en ecuaciones). Además, el modelo propuesto en este estudio, no se consideran poblaciones homogéneas, sino

heterogéneas, incluyendo trenes del STC, usuarios del metro, hospitales y estaciones de bomberos.

Se justifica el no utilizar simulación de dinámica de sistemas (además de no conocer el comportamiento agregado del sistema complejo) debido a que no se está lidiando con agentes homogéneos, situación en la cual a menudo se modelan mejor utilizando una solución agregada como la teoría del campo medio o el modelado de dinámica de sistemas. De acuerdo con Wilensky & Rand (2015) el modelado de dinámica de sistemas requiere la creación de un "stock" separado para cada grupo de agentes con diferentes propiedades y, cuando el espacio de propiedades es grande, esto se vuelve difícil de construir, rastrear e integrar. ABM, por otro lado, solo requiere que se especifique cómo se definen las propiedades de los agentes y no que se realice un seguimiento de todos los tipos de agentes posibles, lo que proporciona una descripción más concisa de un sistema complejo. Wilensky & Rand, 2015) menciona que la rica descripción del medio ambiente y la geografía que conlleva. La modelación basada en agentes permite generar información más detallada. Esto permite que la modelación basada en agentes genere patrones espaciales de resultados en lugar de resultados agregados espacialmente homogéneos.

En conclusión, se justifica el uso de la modelación basada en agentes debido a cuatro razones: la primera es que se considera la movilidad de la CDMX como un sistema complejo del cual no se conoce su comportamiento agregado (requisito para utilizar dinámica de sistemas). Segunda razón, se identifican y consideran agentes heterogéneos y sus interacciones (trenes, hospitales y estaciones de bomberos). Tercera razón es que la modelación basada en agentes permite describir el comportamiento del sistema complejo en estudio de una manera natural y fácil de comprender. Finalmente, con la modelación basada en agentes se puede considerar el movimiento en el espacio (coordenadas XY) y tiempo, lo cual no se puede considerar con entidades (Simulación de Eventos Discretos) o procesos (Dinámica de Sistemas).

3.2. JUSTIFICACIÓN DEL USO DEL SOFTWARE NETLOGOTM

- Se trata de un software de código abierto.
- Plataformas de ejecución: Windows, MAC, LINUX, UNIX.
- El ambiente virtual está basado en JAVA.
- Permite visualizar modelos en 2D y 3D.
- Gratuito.

Además de las ventajas técnicas descritas arriba, el software NetLogo™ permite programar agentes con metas simples y cargar el “layout” basado en sistemas de información geográfica, requisitos necesarios para el desarrollo de este estudio.

Otra de las ventajas del uso del software NetLogo™ es que es fácil de aprender, es decir, su nivel de acceso al lenguaje es bastante bajo, sin embargo, esto no excluye que se puedan lograr modelos de simulación sofisticados, justamente los autores del software lo diseñaron con la mentalidad “low threshold, high ceiling”. Adicionalmente, es el programa para modelación basada en agentes más utilizado, junto con Swarm™ desarrollado por el Instituto de Santa Fe, Repast™ desarrollado por el Laboratorio Nacional de Argonne y MASON™ desarrollado por la Universidad George Mason.

También es posible escribir un modelo basado en agentes en cualquier lenguaje de programación, como podría ser Python cuya ventaja es que el tiempo para ejecutar el modelo puede ser más rápido, sin embargo, a menudo el tiempo para el ciclo de vida de desarrollo del modelo será considerablemente más lento.

Finalmente, NetLogo™ también tiene la gran ventaja sobre el pseudocódigo de ser ejecutable, por lo que se puede ejecutar y probar los modelos fácil y rápidamente, es decir, NetLogo™ es tanto un lenguaje de modelación como un entorno integrado de desarrollo, lo que permite, como se mencionó previamente, que sea más fácil de construir y probar el modelo. Además, hay una gran cantidad de modelos basados en agentes escritos en NetLogo™ en una amplia variedad de dominios y una plétórica cantidad de ejemplos que ayudan a desarrollar ejemplo propios, como una comunidad activa de usuarios que pueden ayudar en caso de dudas.

3.3. SOFTWARE DE SIMULACIÓN NETLOGO™

A continuación se hace una descripción del software que se utilizó para desarrollar el sistema basado en simulación-optimización que se propone en el trabajo de investigación presente.

“NetLogo™ es un entorno de modelado programable de múltiples agentes. Es utilizado por cientos de miles de estudiantes, profesores e investigadores en todo el mundo. También potencia las simulaciones participativas de HubNet. Está escrito por Uri Wilensky y desarrollado en CCL. La parte ‘Logo’ se debe a que NetLogo™ es un dialecto del lenguaje Logo. ‘Net’ pretende evocar la naturaleza descentralizada e interconectada de los fenómenos que puede modelar con NetLogo, incluidos los fenómenos de red. También se refiere a HubNet, el entorno de simulación participativa multiusuario incluido en NetLogo™”

“NetLogo™ fue creado por primera vez en 1999 por Uri Wilensky en el Centro de Aprendizaje Conectado y Modelado Basado en Computadora, luego en la Universidad de Tufts en el área de Boston. NetLogo™ surgió de StarLogoT, que fue escrito por Wilensky en 1997. En 2000, la CCL se mudó

a la Universidad Northwestern, en el área de Chicago. NetLogo™ 1.0 salió en 2002, 2.0 en 2003, 3.0 en 2005, 4.0 en 2007, 4.1 en 2009 y 5.0 en 2012"

(Wilensky, U. 1999.)

La versión que se utiliza para desarrollar el modelo de simulación es la 6.2.0, lanzada el 22 de diciembre del 2020.

"NetLogo™ está escrito principalmente en Scala, con algunas partes en Java. (El código Scala se compila en código de bytes de Java y es completamente interoperable con Java y otros lenguajes JVM)." (Wilensky, U. 1999).

3.4. TRATAMIENTO DE MAPAS EN EL SOFTWARE QGIS™

Se realiza un tratamiento de los conjuntos de datos del metro de la Ciudad de México en el software QGIS™, previo a cargarlos en el software NetLogo™ esto debido a que los datos del metro, obtenidos del portal de datos abiertos del gobierno de la Ciudad de México, agrupan todas las estaciones en una misma capa así como a todas las líneas del metro en otra única capa, esto no es funcional para propósitos del modelo por lo que se tienen que separar cada conjunto de estaciones pertenecientes a una línea en una capa, además de separar las líneas en distintas capas, dando como resultado doce capas de estaciones del metro y otras doce correspondientes a cada línea (Sistema de Transporte Colectivo STC, 12 enero 2021).

Del mismo modo, se revisó el conjunto de datos de las avenidas principales, avenidas secundarias (Secretaría de Movilidad, 28 enero 2021), hospitales y centros de salud, así como el de los hospitales del ISSTE (Secretaría de Salud, 12 enero 2021), datos que fueron obtenidos del mismo portal de datos abiertos del gobierno de la CDMX. Sin embargo, para el caso de las estaciones de bomberos, no se encontró un conjunto de datos que fuera de acceso libre, por lo que se realizó la capa en QGIS™ a partir de las coordenadas tomadas de Google Maps.

En la figura 3.2, se muestra el mapa realizado con todas las capas que se utilizarán en el modelo de NetLogo™. Los datos SIG obtenidos en el Portal de Datos Abiertos del Gobierno de la Ciudad de México, así como el mapa que se presenta en la figura 3.2, están disponibles en la liga que se presenta en la sección "3.14 Replicación o Reproducción del Modelo" de este trabajo.

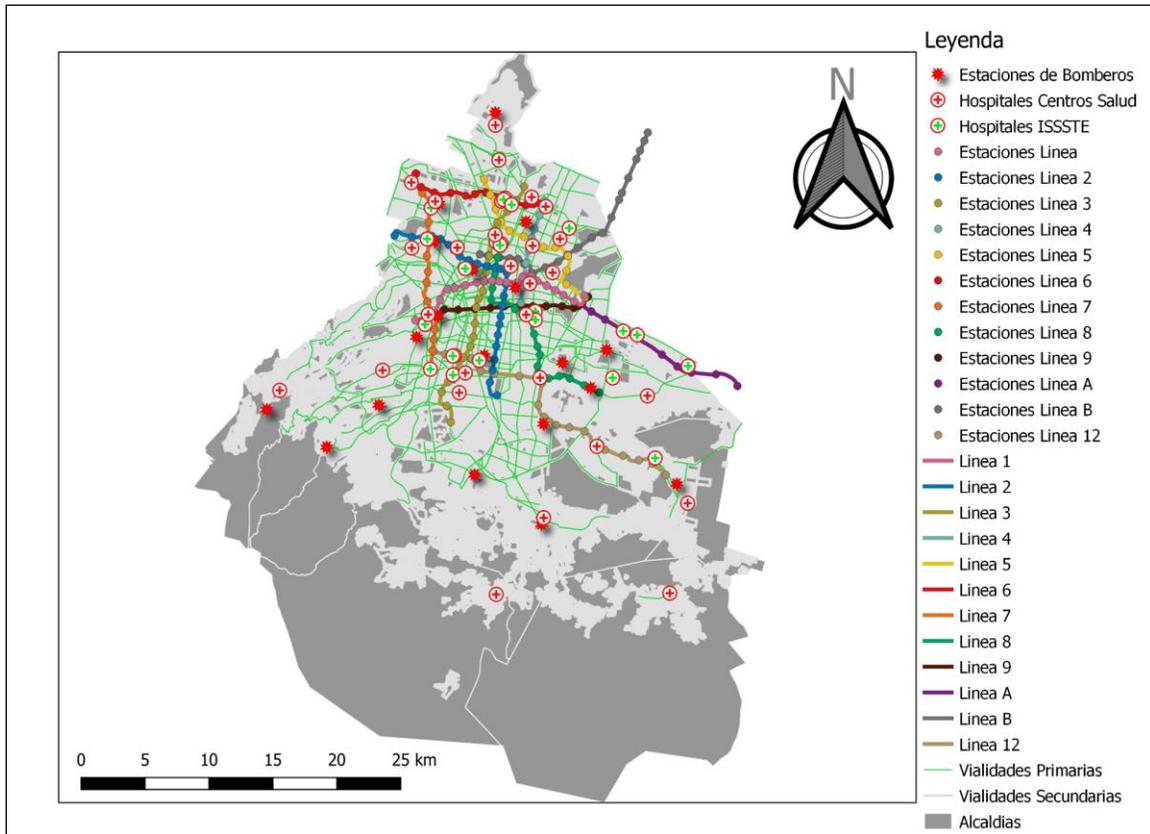


Figura 3.1 Capas de datos geográficos.
Fuente: Elaboración propia (2022).

3.5. *MODELO CONCEPTUAL*

3.5.1. *Modelo conceptual inicial*

Parte de la metodología de modelación utilizada (modelación iterativa) consiste en revisar y modificar el modelo conceptual conforme se realiza la implementación del modelo conceptual, esto con el propósito de que tanto el modelo conceptual como el modelo computacional evolucionen en conjunto y de la misma forma se logre una mejor verificación y validación.

Se observa un mapa de la Ciudad de México, figura 3.2, con las vialidades primarias y secundarias, la red del metro y la ubicación de hospitales y estaciones de bomberos. Sobre la red del metro, se desplazan trenes en las diversas líneas y de acuerdo con una probabilidad "X", un tren sufre del incidente denominado *cortocircuito por llanta ponchada*. En el momento en el que el tren se detiene a causa del incidente, el operador del tren llama a las direcciones correspondientes y pide auxilio a los servicios de emergencia en caso de que algún pasajero dentro del tren haya sufrido alguna lesión a causa del incidente y/o se haya producido

algún incendio en la unidad, de esta forma se generan cuatro escenarios: el primer escenario es que suceda el incidente y ningún pasajero hay sufrido alguna lesión y tampoco haya riesgo de incendio, por lo tanto el operador sólo notifica a las direcciones del STC; el segundo escenario es que suceda el incidente y a causa de esto algún pasajero resultó herido, sin embargo, no hay riesgo de incendio por lo tanto el operador avisa a las direcciones del STC y estas a su vez notifican al hospital más cercano para que envíe una ambulancia; el tercer escenario es que a causa del incidente exista riesgo de incendio, por lo tanto el operador notifica a las direcciones del STC y estas notifican a la estación de bomberos más cercana; finalmente el cuarto escenario es que a causa del incidente haya personas con lesiones y exista riesgo de incendio por lo que el operador notifica a las direcciones del STC y estas avisan al hospital y estación de bomberos más cercanos. En cualquiera de los casos 2, 3 y 4 se identifica el hospital y/o estación de bomberos más cercanos al punto donde se encuentra el tren detenido por el incidente y se verifica si cuentan con recursos o unidades disponibles para atender la emergencia, en caso de que no se cuenten con recursos se localiza la siguiente estación de bomberos y/o hospital más cercano, en caso de que sí se cuenten con recursos se envía una ambulancia y/o un camión de bomberos al lugar del incidente, siendo el punto inicial el hospital o la estación de bomberos y el punto final donde está detenido el tren. Tanto el camión de bomberos como la ambulancia deben de tomar la ruta más corta. En caso de que el hospital o la estación de bomberos no se encuentren adyacentes a una vialidad primaria el camión de bomberos o la ambulancia se desplazará sobre una vialidad secundaria hasta la vialidad primaria más cercana para continuar con la ruta más corta sobre la vialidad primaria.

El nivel al que se trabaja el presente modelo de simulación basado en agentes es en un enfoque mesoscópico, definido así porque no se estiman las interacciones individuales entre coches y ambulancias o camiones de bomberos como sería en un modelo microscópico sino que se pretende establecer sugerencias de rutas que deben de tomar los servicios de emergencia, del mismo modo no se conceptualiza como un enfoque macroscópico debido a que no se abordan las características del flujo de tránsito de manera global. Se considera la red vial de la Ciudad de México, en particular se toman en cuenta las vialidades primarias definidas por la SEMOVI (Secretaría de Movilidad), como son: Anillo Periférico, Circuito Interior, Ejes Viales y Calzadas.

Son cuatro tipos de agentes los que se consideran en el modelo, el primer tipo de agentes son los agentes tipo ruta y tiene dos subtipos los de red del metro y los de las vialidades primarias sobre los que viajan los demás agentes y vialidades secundarias que conectan a las primarias con los hospitales y ambulancias; sobre estos agentes tipo ruta es que el resto de los agentes, metro, ambulancia y camión de bomberos se pueden desplazar. El segundo tipo de agentes son los trenes de metro, los cuales simulan tener el incidente durante su trayecto en las líneas del metro. El tercer tipo de agentes son las ambulancias que se generan en los puntos geográficos de los hospitales públicos y cuyo destino son los puntos donde se genera algún incidente de *cortocircuito por llanta ponchada* en cualquier estación o

segmento de línea del metro. Similar al tercer tipo, el cuarto tipo de agentes son los camiones de bomberos que se generan en las estaciones de bomberos y comparten el mismo destino que las ambulancias.

Tanto los agentes tipo ambulancia como los agentes tipo camión de bomberos, comparten el propósito de tomar la ruta más corta desde su punto de generación hasta el lugar donde se genera el incidente estudiado. En el caso de los agentes tipo tren, su propósito es recorrer las líneas del metro y ocasionalmente simular el incidente de *cortocircuito por llanta ponchada*, lo que activa el proceso para que los otros agentes se desplacen hacia este agente.

El espacio que tienen los agentes tipo tren, está definido por las estaciones y segmentos de línea del metro. El espacio que comparten los agentes tipo ambulancia y camiones de bomberos son las vialidades primarias de la CDMX; para conectar los hospitales y estaciones de bomberos con las vialidades primarias, se consideran vialidades secundarias, sin embargo, el desplazamiento de los agentes tipo ambulancia y camión de bomberos, debe de ser sobre las vialidades primarias.

3.5.2. *Modelo conceptual final*

En el modelo conceptual final se eliminan los elementos que consideraban el movimiento e interacción de coches en las vías principales y secundarias de la CDMX esto para que el modelo se enfocara en el estudio de la evaluación de la localización de los servicios de emergencia como hospitales y estaciones de bomberos a cualquier punto de incidente que se pueda generar dentro la red del metro mediante la obtención de rutas más cortas. Del mismo modo no se realiza el movimiento de las ambulancias o camiones de bomberos sobre la ruta más corta, sólo se obtiene la ruta más corta desde el lugar del incidente al hospital o estación de bomberos más cercana que cuente con recursos, esto debido a que se prioriza obtener la mayor cantidad de incidentes para obtener los promedios de cada línea y reducir recursos computacionales. También no se dibuja con agentes la red de las vialidades secundarias para ahorrar recursos computacionales y por lo tanto sólo se considera la red de avenidas principales sobre la cual se calcula la ruta más corta.

Como se mencionó previamente las interacciones de coches, ambulancias y camiones de bomberos no se toman en cuenta por lo que los agentes tipo camiones de bomberos, ambulancias y coches se descartan. Los agentes que permanecen son los tipo: metro, tipo ruta de avenida principal, tipo hospital, tipo y estaciones de bomberos. Los agentes tipo metro tienen las propiedades de pasajeros, probabilidad de incendio, probabilidad de heridos, probabilidad de cortocircuito, dirección y línea. Los agentes tipo ruta ya sean de avenidas principales o de líneas del metro tienen la propiedad de reconocer a sus vecinos para formar la red. Los agentes tipo hospital y estaciones de bomberos, tienen la propiedad de recursos, lo que les permite establecer si se cuentan con los recursos suficientes para establecer la ruta más corta hacia el incidente o en caso

de que no cuenten con recursos suficientes se descarta y sólo se toma en cuenta aquél que sea el más cercano tanto de hospitales como de estaciones de bomberos.

Además en este modelo de simulación, las interacciones que van a tener los agentes entre sí, consisten en que los agentes de tipo tren generan un incidente y activan a los agentes de tipo hospital y estación de bomberos para revisar si cuentan con recursos suficientes para atender el incidente. No se toman en cuenta las interacciones que se generan entre las ambulancias y el tránsito de la ciudad, es decir, no hay relación entre ambulancias y coches y/o camiones de bomberos con los coches que transitan las vialidades.

Es importante establecer que todas las probabilidades involucradas en el modelo no son dependientes de eventos pasados. En cada tic se obtiene una nueva probabilidad ya sea de incendio, heridos, incidente de cortocircuito y la probabilidad de que haya recursos en hospitales y estaciones de bomberos, por lo que los hospitales y estaciones de bomberos disponibles serán diferentes en cada tic.

El modelo conceptual queda de la siguiente forma, figura 3.2:

“Se observa un mapa de la Ciudad de México, figura 3.1, con las vialidades primarias y secundarias, la red del metro y la ubicación de hospitales y estaciones de bomberos. Sobre la red del metro, se desplazan trenes en las diversas líneas y de acuerdo con una probabilidad "X", un tren sufre del incidente denominado *cortocircuito por llanta ponchada*. En el momento en el que el tren se detiene a causa del incidente, el operador del tren llama a las instancias correspondientes y pide auxilio a los servicios de emergencia en caso de que algún pasajero dentro del tren haya sufrido alguna lesión a causa del incidente y/o se haya producido algún incendio en la unidad, de esta forma se generan cuatro escenarios: el primer escenario es que suceda el incidente y ningún pasajero hay sufrido alguna lesión y tampoco haya riesgo de incendio, por lo tanto el operador sólo notifica a las autoridades del STC; el segundo escenario es que suceda el incidente y a causa de esto algún pasajero resultó herido, sin embargo, no hay riesgo de incendio por lo tanto el operador avisa a las autoridades del STC y estas a su vez notifican al hospital más cercano para que envíe una ambulancia; el tercer escenario es que a causa del incidente exista riesgo de incendio, por lo tanto el operador notifica a las autoridades del STC y estas notifican a la estación de bomberos más cercana; finalmente el cuarto escenario es que a causa del incidente haya personas con lesiones y exista riesgo de incendio por lo que el operador notifica a las autoridades del STC y estas avisan al hospital y a la estación de bomberos más cercanos. En cualquiera de los casos 2, 3 y 4 se identifica el hospital y/o estación de bomberos más cercanos al punto donde se encuentra el tren detenido por el incidente y se verifica si cuentan con recursos o unidades disponibles para atender la emergencia, en caso de que no se cuenten con recursos se localiza la siguiente estación de bomberos y/o hospital más cercano. Se obtienen la

distancia de la ruta más corta y se guarda para sacar el promedio con las siguientes rutas sucesivas.”

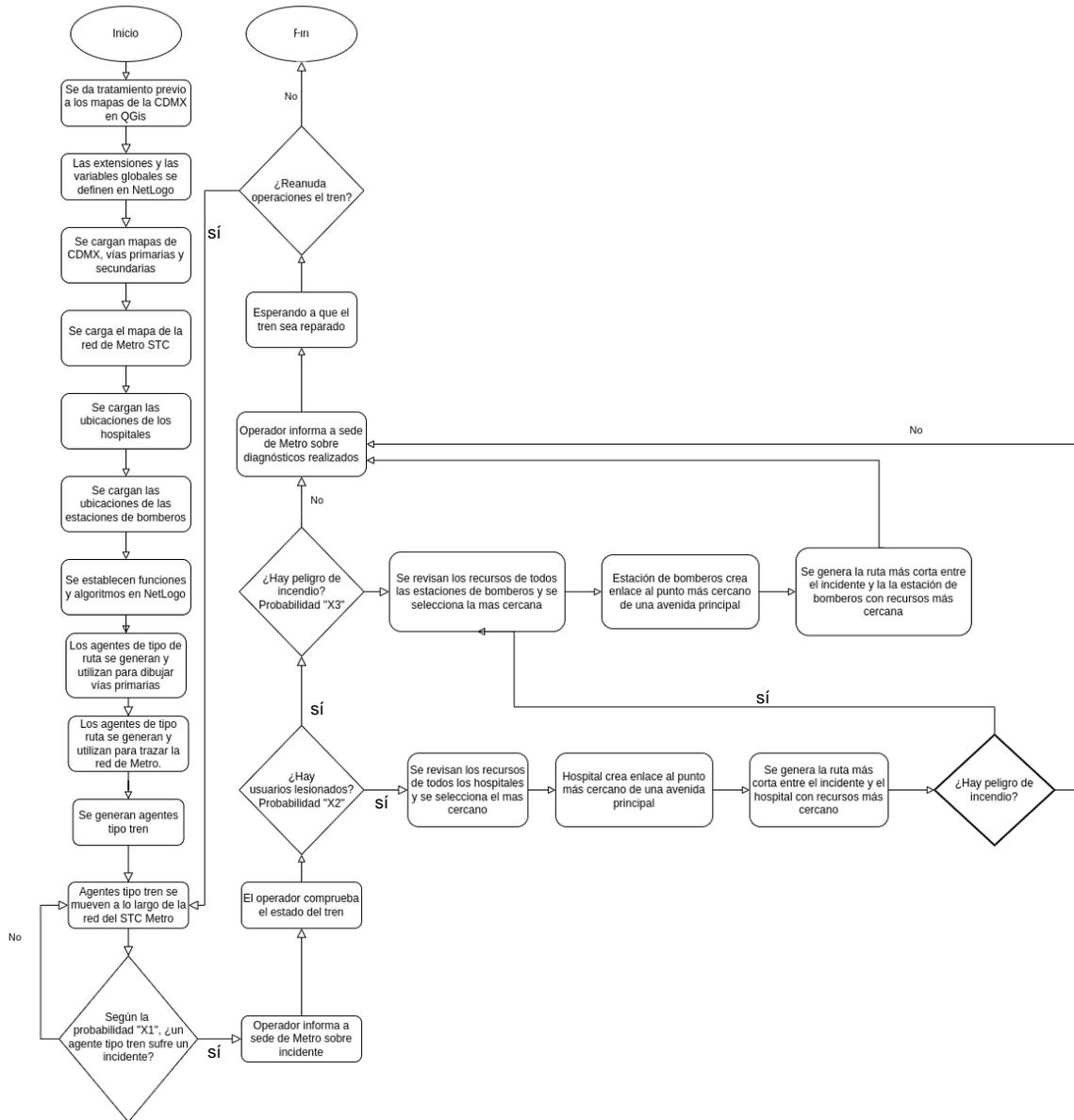


Figura 3.2 Diagrama de flujo del modelo de simulación basado en agentes.
Fuente: Elaboración propia (2022)

3.6. ALGORITMO DEL MODELO

De acuerdo con (Wilensky & Rand, 2015, 79) cuando se trabaja con modelos basados en agentes, puede ser útil utilizar un formulario basado en texto para describir explícitamente las reglas del modelo. Este formato de pseudocódigo debe tener dos secciones: “Inicializar”, que describe cómo funcionan las condiciones iniciales del modelo, y “En cada tic”, que describe el comportamiento

de los agentes en cada tic del reloj del software NetLogo™. Para los modelos basados en agentes, se recomienda escribir el texto desde el punto de vista de los agentes, especialmente en la sección "En cada tic".

3.6.1. *Pseudocódigo general*

Un pseudocódigo se puede definir como el paso previo a la implementación en un lenguaje de programación, ya sea: Python, C, Java, JavaScript, etc., es decir, tiene un orden en las instrucciones u operaciones, sin embargo, no tiene la sintaxis característica del lenguaje de programación seleccionado. Un código de programación, es una sección de un programa computacional que sigue las reglas y la sintaxis de cada lenguaje de programación, es por esto que una serie de instrucciones descritas en un pseudocódigo puede adaptarse a varios lenguajes de programación, sin embargo, la misma serie de instrucciones tendrá diferente forma y estructura en cada lenguaje de acuerdo con las características del lenguaje de programación

Aquí se presenta el pseudocódigo general original y final del modelo computacional. Este pseudocódigo corresponde al diagrama de flujo de la figura 3.2.

- 1) Se carga un mapa de la Ciudad de México con las vialidades primarias y secundarias.
- 2) Se carga la red del metro.
- 3) Se cargan las ubicaciones de hospitales.
- 4) Se cargan las ubicaciones de estaciones de bomberos.
- 5) Se dibuja con agentes la red del metro.
- 6) Sobre la red de metro, se desplazan trenes en las diversas líneas.
- 7) De acuerdo con una probabilidad "X1", cualquier tren sufre del incidente denominado *cortocircuito por llanta ponchada*.
- 8) El tren se detiene en el nodo actual del incidente y el operador del tren llama a las direcciones correspondientes del STC.
- 9) El operador revisa el estado del tren. Primero verifica si hay heridos, si hay heridos luego verifica si hay riesgo de incendio. Si no hay heridos luego verifica si hay riesgo de incendio
- 10) Si hay usuarios lesionados de acuerdo con una probabilidad "X2", llama al hospital más cercano al tren, en caso contrario, notifica a las direcciones que no hay lesionados.
- 11) En caso de heridos, se verifican los recursos de todos los hospitales, aquellos que no cuenten con recursos de acuerdo con una probabilidad del 50% se descartan y se selecciona el más cercano.
- 12) Si el hospital no está adyacente a una vialidad primaria, el agente hospital crea un enlace al punto más cercano de la avenida principal.
- 13) Si existe riesgo de incendio de acuerdo con una probabilidad "X3" llama a la estación de bomberos más cercana, en caso contrario, notifica a las direcciones que no hay riesgo de incendio.

- 14) En caso de que llame a la estación de bomberos se verifican los recursos de todas las estaciones de bomberos, aquellas que no cuenten con recursos suficientes de acuerdo con una probabilidad del 50% se descartan y se selecciona la más cercana al incidente.
- 15) Si la estación de bomberos no está adyacente a una vialidad primaria, el agente estación de bomberos crea un enlace al punto más cercano de la avenida principal.
- 16) Si no hay lesionados o riesgo de incendio, se notifica a las direcciones correspondientes del STC y se repara el tren.
- 17) En cuanto el tren haya sido reparado, se reanuda la operación del tren y se calculan nuevamente todas las probabilidades con cada tic.

3.6.2. *Pseudocódigo “inicializar”*

Se presenta el pseudocódigo del modelo general, es decir, el modelo de comportamiento agrupado con todas las líneas del metro, sin embargo, los sub-modelos para línea del metro tienen un pseudocódigo similar. Se mantuvo constante la mayor parte de la implementación.

To “setup” (inicializar):

- 1) Se cargan los datos del mapa de la Ciudad de México con las vialidades primarias y secundarias.
- 2) Se cargan los datos de la red del metro.
- 3) Se cargan los datos de las ubicaciones de hospitales.
- 4) Se cargan los datos de las ubicaciones de estaciones de bomberos.
- 5) Se dibuja con agentes y enlaces direccionados cada sentido de cada línea del metro.
- 6) Se dibuja con agentes la red del metro.
- 7) Se dibuja con agentes las vialidades primarias
- 8) Si se desea se carga el GIS de las vialidades secundarias
- 9) Se dibuja con agentes las vialidades secundarias
- 10) Se crean un número par de trenes definido por un slider, la mitad de los trenes se generan en un sentido de la línea y la otra mitad en el otro sentido
- 11) Crear NÚMERO de trenes, donde NÚMERO está definido por un slider en la interface
- 12) Cada tren se mueve a un punto aleatorio de cada línea del metro.
- 13) Se separan los agentes tipo tren para que no haya dos en un mismo punto.
- 14) Se crean los agentes tipo hospital.
- 15) Se crean los agentes tipo estaciones de bomberos.

3.6.3. *Pseudocódigo para cada “tic”*

Se presenta el pseudocódigo de movimiento o para cada tic cómo se comportan los agentes. En el pseudocódigo original se consideraba el movimiento de coches, sin embargo, debido al enfoque y ahorro de recursos, se retiró, las partes

originales se muestran en azul. Este pseudocódigo corresponde al diagrama de flujo de la (figura 3.7).

- 1) Generación aleatoria de probabilidad de cortocircuito
- 2) Generación aleatoria de pasajeros para cada tren
- 3) Generación aleatoria de probabilidad de heridos
- 4) Generación aleatoria de probabilidad de incendio
- 5) Función general de movimiento de trenes

3.7. DIAGRAMA DEL MOVIMIENTO DE TRENES

Debido a la complejidad e importancia de la rutina de programación para el movimiento de los trenes, en la figura 3.3 se presenta el diagrama de flujo que representa este movimiento. Este diagrama de flujo puede considerarse dentro del diagrama de flujo de la figura 3.2 cada que se mueven los trenes.

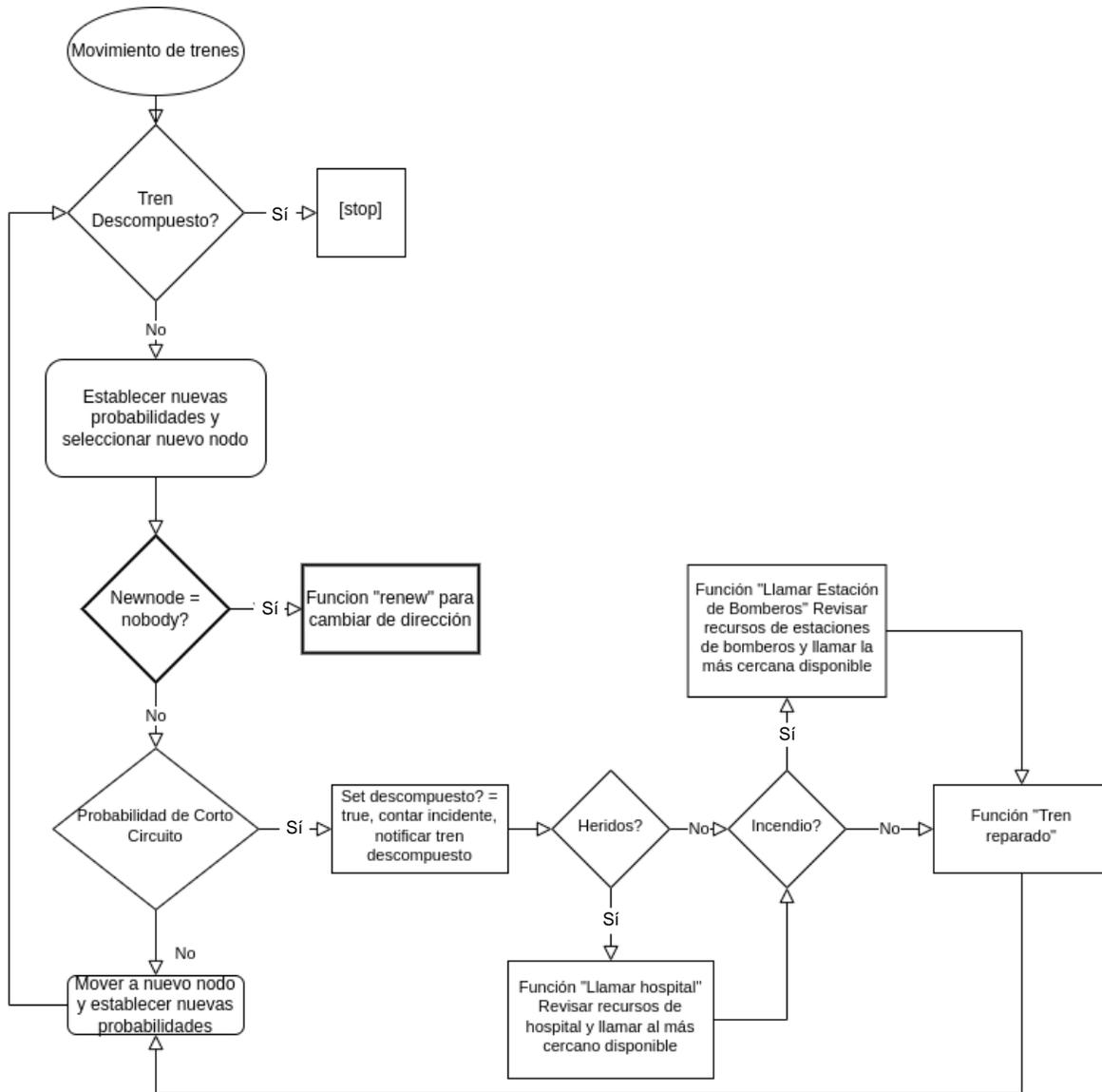


Figura 3.3 Diagrama de flujo para el movimiento de trenes.
Fuente: Elaboración propia (2022).

3.8. MODELO COMPUTACIONAL

3.8.1. Explicación de la interfaz del modelo

En esta sección, se presenta la explicación de la interfaz del modelo. Como se puede observar en la (figura 3.8) la interfaz cuenta con varios botones, *switches* o interruptores, *choosers* o seleccionadores, *sliders* deslizador, contadores y gráficas que a continuación se describe el funcionamiento de cada grupo indicado con los colores de la (figura 3.8). En el Anexo A, se describe el uso de cada uno de estos elementos de NetLogo™.

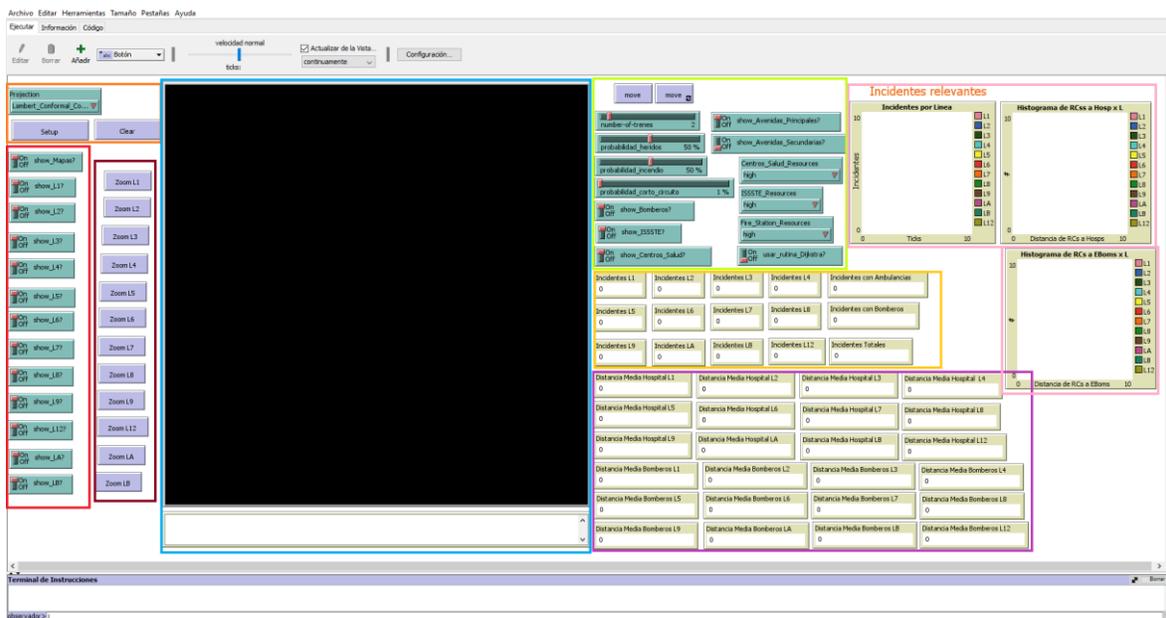


Figura 3.4 Interfaz del modelo.
Fuente: Elaboración propia (2022).

En la (figura 3.9) se observa primero en la parte superior izquierda el seleccionador de la proyección cartográfica que sea quiera utilizar, ya sea de Cónica de Lambert, la WGS84 (sistema geodésico de coordenadas) o la *US Orthographic*. Debajo del seleccionador de las proyecciones, se tienen dos botones, el de la izquierda “setup” o inicializar permite establecer las condiciones iniciales para el modelo de comportamiento agregado o modelo general, el botón de la derecha permite borrar todas las condiciones generales, limpiar la pantalla de visualización y regresar a cero los tics.



Figura 3.5 Botones para configurar las condiciones iniciales del modelo.
Fuente: Elaboración propia (2022).

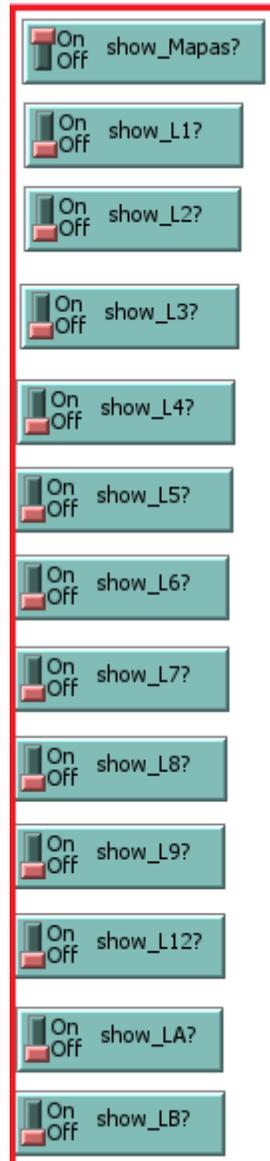


Figura 3.6 “Switches” para habilitar las distintas líneas de la red del metro.
Fuente: Elaboración propia (2022).

En la (figura 3.10) se encuentran los interruptores que permiten visualizar las distintas líneas del metro en el modelo de comportamiento agregado o general, así

como visualizar el mapa de las alcaldías de la CDMX. Se pueden tener todos activados o algunos activados de los interruptores de las líneas del metro.

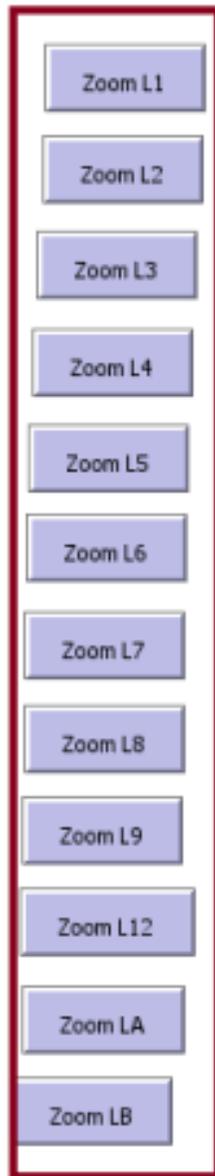


Figura 3.7 Botones que permiten visualizar sub-modelos de cada línea de la red del metro.

Fuente: Elaboración propia (2022).

La sección de botones de la (figura 3.11) es muy importante porque cada botón establece las condiciones iniciales de cada sub-modelo de cada línea de la red del metro, es decir, si sólo se desea analizar la línea 1 del metro se da clic en el botón “Zoom L1” y se generan las condiciones iniciales para esa línea considerando los bordes del “mundo” de NetLogo™, este límite espacial abarca las estaciones de la línea, es decir de terminal a terminal son los bordes del “mundo” y todas las avenidas principales que se captan en ese cuadro.

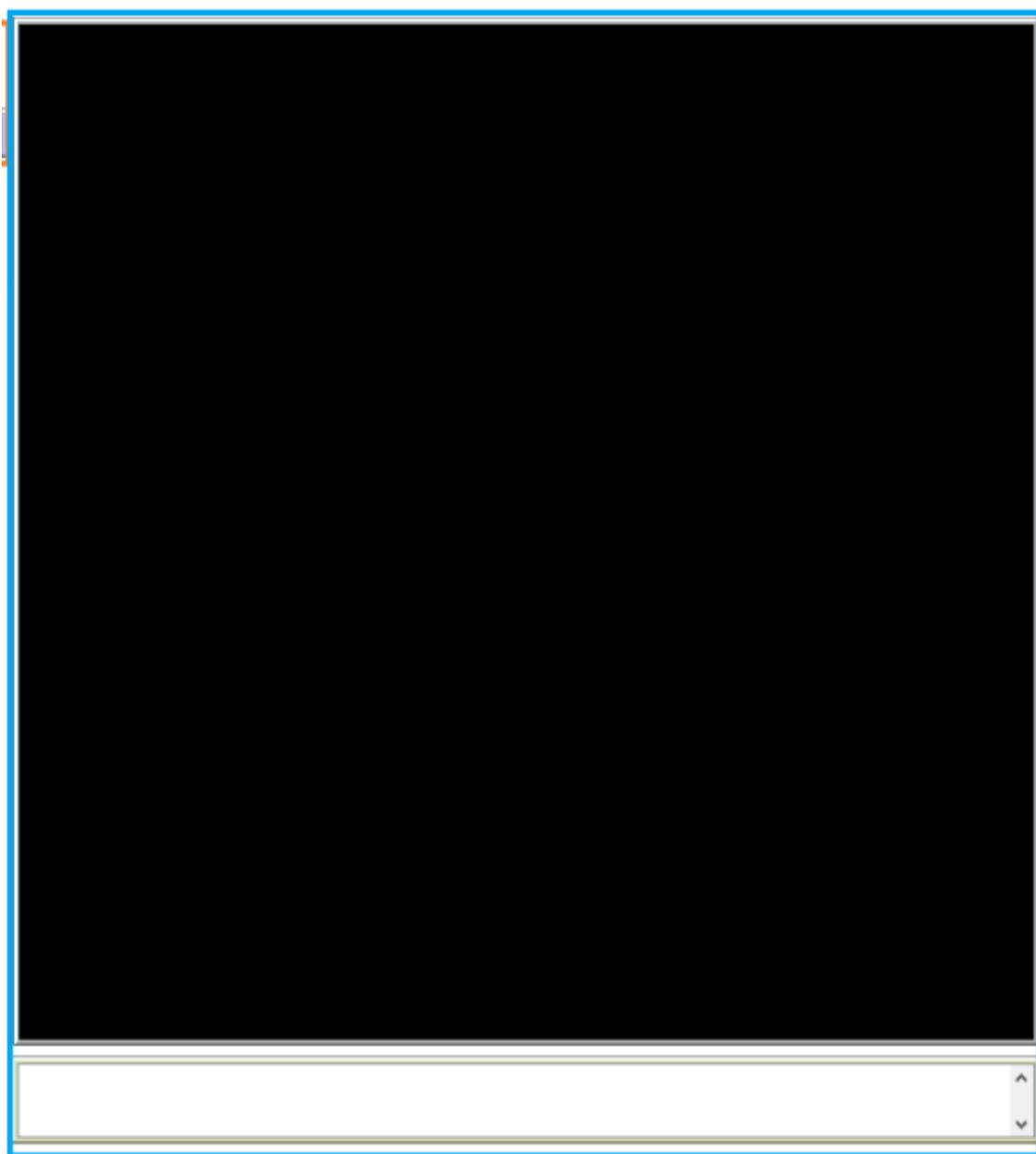


Figura 3.8 Pantalla de visualización de los mapas y caja de reportes.
Fuente: Elaboración propia (2022).

En la (figura 3.12) se encuentra la pantalla donde se generan todos los agentes, enlaces, parches y se cargan los mapas de GIS, abajo se encuentra una caja donde se reportan los incidentes; de qué línea son, en qué dirección viajaba el tren, si requiere llamar al hospital o estación de bomberos y cuando ha sido reparado el tren.

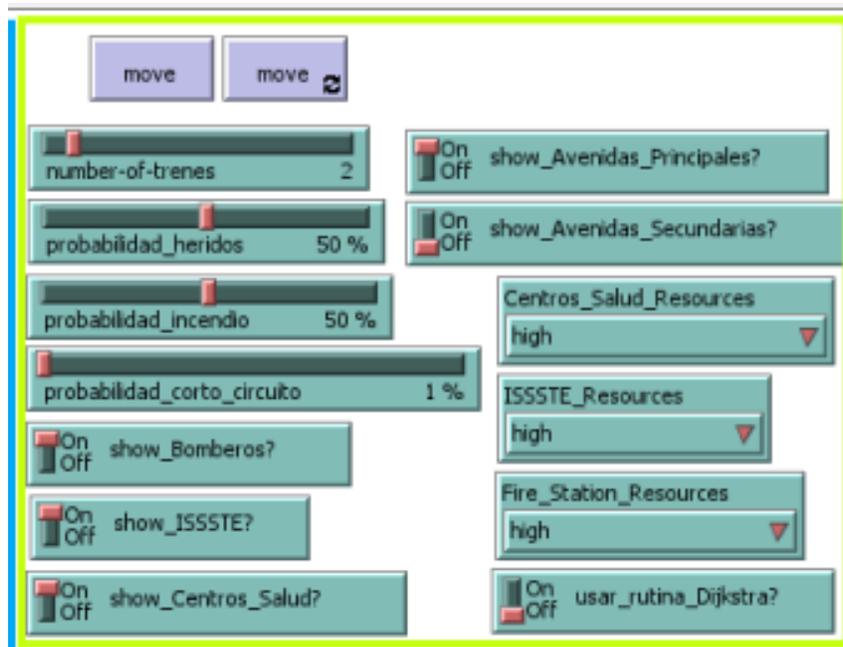


Figura 3.9 Sliders y switches de probabilidad y visualización de hospitales y estaciones de bomberos, choosers de recursos y botones de movimiento.

Fuente: Elaboración propia (2022).

La sección de la (figura 3.13) contiene los botones de “move” o movimiento para cada tic y “move” de tics continuos, debajo se encuentra un “slider” para configurar el número de trenes que tendrá cada línea y aumenta en pares debido a que la mitad de trenes estarán en una dirección y la otra mitad en la otra dirección. También se encuentran los deslizador que configuran las probabilidades que gobiernan el incidente de cortocircuito y si se requieren los servicios de hospitales y estaciones de bomberos. Así mismo, también se encuentran los interruptores que habilitan o deshabilitan la visualización de las avenidas principales, avenidas secundarias estaciones de bomberos y hospitales. En caso de estar deshabilitados las probabilidades correspondientes de heridos e incendio, para el caso de la visualización de ISSSTE-Centros de Salud y Bomberos, deberán estar en cero o el modelo indicará que se requiere habilitar la visualización para que se generen los incidentes. También se encuentran los seleccionadores que permiten escoger la configuración de los recursos en cada tic de los hospitales y estaciones de bomberos, si escoge “high” siempre tendrán recursos y si se escoge “low” tendrán un 50% de tener o no recursos. Finalmente, hay un último interruptor que permite la habilitación de la rutina Dijkstra adaptada de otro modelo (Gil Alvaro, 2012) a este modelo.

Incidentes L1 0	Incidentes L2 0	Incidentes L3 0	Incidentes L4 0	Incidentes con Ambulancias 0
Incidentes L5 0	Incidentes L6 0	Incidentes L7 0	Incidentes L8 0	Incidentes con Bomberos 0
Incidentes L9 0	Incidentes LA 0	Incidentes LB 0	Incidentes L12 0	Incidentes Totales 0

Figura 3.10 Contadores de incidentes para cada línea del metro y contadores de incidentes que involucren Ambulancias y Camiones de Bomberos.
Fuente: Elaboración propia (2022).

En la (figura 3.14) se presentan los contadores de incidentes para cada línea del metro, incluye también los totales que involucren llamar ambulancias/hospitales y el caso similar para las estaciones de bomberos.

Distancia Media Hospital L1 0	Distancia Media Hospital L2 0	Distancia Media Hospital L3 0	Distancia Media Hospital L4 0	0 0
Distancia Media Hospital L5 0	Distancia Media Hospital L6 0	Distancia Media Hospital L7 0	Distancia Media Hospital L8 0	
Distancia Media Hospital L9 0	Distancia Media Hospital LA 0	Distancia Media Hospital LB 0	Distancia Media Hospital L12 0	
Distancia Media Bomberos L1 0	Distancia Media Bomberos L2 0	Distancia Media Bomberos L3 0	Distancia Media Bomberos L4 0	
Distancia Media Bomberos L5 0	Distancia Media Bomberos L6 0	Distancia Media Bomberos L7 0	Distancia Media Bomberos L8 0	
Distancia Media Bomberos L9 0	Distancia Media Bomberos LA 0	Distancia Media Bomberos LB 0	Distancia Media Bomberos L12 0	

Figura 3.11 Contadores de distancias media para cada línea hacia el hospital o estación de bomberos más cercana.
Fuente: Elaboración propia (2022).

De la (figura 3.15) se observan los contadores de las distancias promedio de las rutas más cortas que se generan desde cualquier incidente en una línea del metro al hospital y/o estación de bomberos más cercana



Figura 3.12 Gráfica de incidentes por línea e histograma de rutas más cortas a hospitales por línea.
Fuente: Elaboración propia (2022).

En el lado izquierdo de la (figura 3.16) se encuentra la gráfica de incidentes que se generan en el tiempo (tics) donde se incluyen todas las líneas del metro. Del lado derecho se encuentra el histograma con las distancias de las rutas más cortas desde el incidente al hospital más cercano para cada línea del metro.

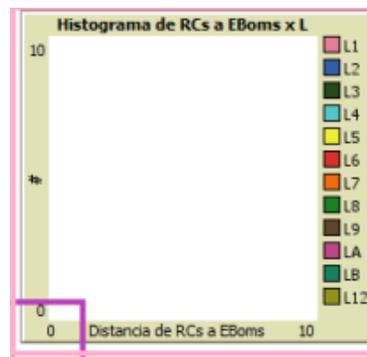


Figura 3.13 Histograma de rutas más cortas a estaciones de bomberos por línea.
Fuente: Elaboración propia (2022).

El histograma de rutas más cortas desde el incidente a la estación de bomberos más cercana que se observa en la (figura 3.17) es complementario al histograma que se describió previamente.

3.8.2. *Explicación general del código*

En esta sección se realiza la explicación de algunas secciones del código desarrollado para el modelo de simulación-optimización planteado. Esto con el fin de ayudar a la verificación y validación del modelo.

```

+ extensions [
directed-link-breed[EnlacesL1Norte EnlaceL1Norte]
directed-link-breed[EnlacesL1Sur EnlaceL1Sur]
directed-link-breed[EnlacesL2Norte EnlaceL2Norte]
directed-link-breed[EnlacesL2Sur EnlaceL2Sur]
directed-link-breed[EnlacesL3Norte EnlaceL3Norte]
directed-link-breed[EnlacesL3Sur EnlaceL3Sur]
directed-link-breed[EnlacesL4Norte EnlaceL4Norte]
directed-link-breed[EnlacesL4Sur EnlaceL4Sur]
directed-link-breed[EnlacesL5Norte EnlaceL5Norte]
directed-link-breed[EnlacesL5Sur EnlaceL5Sur]
directed-link-breed[EnlacesL6Norte EnlaceL6Norte]
directed-link-breed[EnlacesL6Sur EnlaceL6Sur]
directed-link-breed[EnlacesL7Norte EnlaceL7Norte]
directed-link-breed[EnlacesL7Sur EnlaceL7Sur]
directed-link-breed[EnlacesL8Norte EnlaceL8Norte]
directed-link-breed[EnlacesL8Sur EnlaceL8Sur]
directed-link-breed[EnlacesL9Norte EnlaceL9Norte]
directed-link-breed[EnlacesL9Sur EnlaceL9Sur]
directed-link-breed[EnlacesLANorte EnlaceLANorte]
directed-link-breed[EnlacesLASur EnlaceLASur]
directed-link-breed[EnlacesLBNorte EnlaceLBNorte]
directed-link-breed[EnlacesLBSur EnlaceLBSur]
directed-link-breed[EnlacesL12Norte EnlaceL12Norte]
directed-link-breed[EnlacesL12Sur EnlaceL12Sur]
undirected-link-breed[EnlacesAP EnlaceAP]
breed[verticesAP vertexAP]
breed[verticesNL1 vertexNL1]
breed[verticesSL1 vertexSL1]
breed[verticesNL2 vertexNL2]
breed[verticesSL2 vertexSL2]
breed[verticesNL3 vertexNL3]

```

Figura 3.14 Primera sección del código: "breeds".
Fuente: Elaboración propia (2022).

En la primera sección del código se establecen las funciones con las que el modelo trabaja. Las tres extensiones son: 1) “*nw*” que habilita el uso de la extensión Networks para el análisis y construcción de redes, esta extensión es útil porque se utiliza el primitivo “*nw:path-to*” que permite encontrar la ruta más corta entre dos tortugas y lo expresa como una lista de enlaces. 2) La extensión “*gis*” que permite cargar y modificar archivos de sistemas de información geográfica ya sea mediante archivos “.*shp*” (shapefile) o “.*esri*” (ESRI shapefiles) lo que permite trabajar con puntos, líneas y polígonos. 3) La extensión “*table*” que permite usar tablas que son útiles cuando se necesita asociar valores con otros valores. Por ejemplo, puede hacer una tabla de palabras y sus definiciones, esta extensión es utilizada por la rutina del algoritmos Dijkstra adaptada (Gil Álvaro, 2012).

Luego de definir las extensiones se definen las familias o “*breeds*” de los enlaces y agentes con los que se van a trabajar, en este caso se define una familia de enlaces dirigidos para cada dirección de cada línea lo que da un total de 24 familias de enlaces dirigidos; se utilizan enlaces dirigidos para facilitar el movimiento de los trenes aunque también se puede lograr el movimiento con familias de enlaces no-dirigidos, sin embargo, se requiere crear una familia de enlaces ya sea dirigido o no dirigido porque no pueden haber enlaces con nombre (*breeded*) y sin nombre (*unbreeded*) en un mismo espacio. De una forma similar se establecen 24 familias de agentes denominados “*verticesDLX*” donde la D denota la dirección y la X el número de línea, también se generan familias de

agentes para los trenes, y los vértices de las avenidas principales, así como para los hospitales y estaciones de bomberos.

```

globals [
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Propiedades de hospitales y estaciones de bomberos
verticesISSSTE-own[
verticesCENSAL-own[
verticesESTABOM-own[
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Propiedades de agentes que se mueven
trenes-own [
    mynode ;;a vertex. where he begins his trip
    previousnode
    newnode
    probb_CC
    descompuesto?
    heridos?
    incendio?
    pasajeros
    direccion
    linea
    probb_heridos
    probb_incendio
    previouscolor
    ]
ambulancias-own[
camiones_bomberos-own[
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Propiedades de agentes que se usan para dibujar las redes
verticesAP-own [
verticesNL1-own [
verticesSL1-own [
verticesNL2-own [
verticesSL2-own [

```

Figura 3.15 Segunda sección del código: "propiedades".
Fuente: Elaboración propia (2022).

En la figura 3.19 se observa que se encuentran hasta arriba una sección llamada "globals" en esa sección se listan todas las variables a las que se puede acceder durante todo el código, son variables globales, también existen las variables locales pero esas sólo existen dentro de una función, afuera de la función no se reconocen a diferencia de las globales. Luego se identifican las secciones de las propiedades de los agentes con la terminología *FamiliaDeAgentes-own* como en el caso de trenes-own se pueden otorgar propiedades que todos los agentes que pertenecen a la familia tren poseen al generarse; para el caso de los trenes se les asignan: nodo actual, nodos vecinos, descompuesto? (verdadero si está descompuesto falso de lo contrario), probabilidad de cortocircuito, heridos? (verdadero si se hallaron heridos debido al incidente, falso de lo contrario), incendio? (verdadero si existe incendio dentro del tren falso de lo contrario), pasajeros, dirección (en cuanto a terminal), línea (1, 2, 3, ... 12), probabilidad de heridos, probabilidad de incendio, color previo. Las propiedades de los agentes es en parte lo que proporciona validación al modelo ya que permite determinar qué tan parecidos son los agentes con su contraparte en la realidad.

```

.....;Funciones para separar los trenes el moment
to separate-the-trenes-NL1;; turtle procedure
; this procedure is needed so when we click "Setup" we don't end up with any two cars on the same patch
  if any? other trenes-here [
    move-to one-of verticesNL1
    separate-the-trenes-NL1
  ]
end
+ to separate-the-trenes-SL1;; turtle procedure
+ to separate-the-trenes-NL2;; turtle procedure
+ to separate-the-trenes-SL2;; turtle procedure
+ to separate-the-trenes-NL3;; turtle procedure
+ to separate-the-trenes-SL3;; turtle procedure
+ to separate-the-trenes-NL4;; turtle procedure
+ to separate-the-trenes-SL4;; turtle procedure

```

Figura 3.16 Tercera sección del código: "separar trenes".
Fuente: Elaboración propia (2022).

Las funciones de la (figura 3.20) evitan que se generen dos trenes en el mismo nodo esto ayuda a la validación del modelo porque en la vida real dos trenes no pueden ocupar el mismo espacio, sin embargo, en NetLogo™ más de un agente sí puede estar en un mismo "patch" o nodo. Hay una función cada sentido de cada línea con un total de 24 funciones.

```

.....;Función para cargar datos de GIS
to loadDataGis
  gis:load-coordinate-system (word "Data/" projection ".prj")
  ; Load all of our datasets
  set Alcaldias-dataset gis:load-dataset "Data/ALCALDIAS.shp"

  set L1-dataset gis:load-dataset "Data/Linea1.shp"
  set L1-estaciones-dataset gis:load-dataset "Data/Estaciones_Linea1.shp"

  set L2-dataset gis:load-dataset "Data/Linea2.shp"
  set L2-estaciones-dataset gis:load-dataset "Data/Estaciones_Linea2.shp"

  set L3-dataset gis:load-dataset "Data/Linea3.shp"
  set L3-estaciones-dataset gis:load-dataset "Data/Estaciones_Linea3.shp"

  set L4-dataset gis:load-dataset "Data/Linea4.shp"
  set L4-estaciones-dataset gis:load-dataset "Data/Estaciones_Linea4.shp"

  set L5-dataset gis:load-dataset "Data/Linea5.shp"
  set L5-estaciones-dataset gis:load-dataset "Data/Estaciones_Linea5.shp"

```

Figura 3.17 Cuarta sección del código: "cargando datos GIS".
Fuente:Elaboración propia (2022).

La función "loadDataGis" de la figura 3.21 es útil porque permite cargar cuando se necesiten los datos de los sistemas de información geográfica (GIS), los GIS se cargan como una imagen, sin embargo, los agentes no siguen una dirección de la imagen del GIS, para que sigan la dirección de una línea del metro hace falta dibujar la línea del GIS con agentes que funcionen como nodos y formando enlaces entre ellos; esto se logra con familias de agentes y enlaces para cada sentido de cada línea.

```

.....;Funciones para borrar nodos repetidos y nodos que no estén conectados
to delete-duplicatesAP
  ask verticesAP [
    if count verticesAP-here > 1
      ask other verticesAP-here [
        ask myself [create-EnlacesAP-with other [link-neighbors] of myself]
        die]
      ]
  ]
end
to delete-not-connectedAP
  ask verticesAP [set test 0]
  ask one-of verticesAP [set test 1]
  repeat 500 [
    ask verticesAP with [test = 1]
    [ask myneighbors [set test 1]]]
  ask verticesAP with [test = 0][die]
end

```

Figura 3.18 Quinta sección del código: "eliminar duplicados".
Fuente: Elaboración propia (2022).

Las funciones “delete-duplicateAP” y “delete-not-connectedAP” de la figura 3.22 sirven para simplificar la red de agentes dibujada a partir de los datos GIS de las avenidas principales de la Ciudad de México obtenidos del portal de datos abiertos del Gobierno de la Ciudad de México (Secretaría de Movilidad 28 enero 2021), esta red de agentes permite que funcione tanto el algoritmo Dijkstra como la extensión “nw” para encontrar la ruta más corta entre el incidente en el tren y el hospital o estación de bomberos más cercana. Esta simplificación de la red de agentes se realiza para agilizar los cálculos de los algoritmos.

```

.....;Funciones para dibujar redes
to drawAP
  gis:set-drawing-color 27
  gis:draw Vias-principales-dataset 1
  .....;Dibujando red de avenidas principales y colocando coches
  foreach gis:feature-list-of Vias-principales-dataset [ ?1 ->
    foreach gis:vertex-lists-of ?1 ; for the road feature, get the list of vertices
      [ ??1 ->
        let previous-node-pt nobody

        foreach ??1 ; for each vertex in road segment feature
          [ ???1 ->
            let location gis:location-of ???1
            if not empty? location
              {
                ;ifelse any? vertices with [(xcor = item 0 location and ycor = item 1 location) ] ; if there is not a road-vertex here already
                ;[]
                ;[
                create-verticesAP 1
                [set myneighbors n-of 0 turtles ;;empty
                 set xcor item 0 location
                 set ycor item 1 location
                 set size 0.2
                 set shape "circle"
                 set color brown
                 set hidden? true
                ;; create link to previous node
                ifelse previous-node-pt = nobody
                [] ; first vertex in feature
                [create-EnlaceAP-with previous-node-pt] ; create link to previous node
                 set previous-node-pt self]
                ;]
              ] ] ] ]
  delete-duplicatesAP
  delete-not-connectedAP
  ask EnlacesAP [set color gray
  ;set hidden? true
  ]
end

```

Figura 3.19 Sexta sección del código: “Dibujar GIS con agentes”.
Fuente: Elaboración propia (2022).

La función “drawlíneaX” aplica para cada sentido de cada línea y para las avenidas principales, esta rutina fue adaptada del modelo de ruta más corta para la Universidad de George Mason (George Mason University, S.F.) y permite dibujar nodos y enlaces como están expresados en el archivo “.shp”. Esta rutina tiene varias ventajas sobre algunos primitivos de la extensión GIS como también desventajas, entre las ventajas es que no se tuvo problemas para replicar las coordenadas XY de los elementos del GIS (puntos y líneas) para cada agente a diferencia del primitivo de la extensión “gis”: “gis:create-turtles-from-points” la cual permite crear agentes para los elementos del GIS y debido a la naturaleza de los datos GIS utilizados marcaba ciertos errores, sin embargo, este primitivo tiene la ventaja de asignar directamente el nombre a las propiedades que hayan sido nombradas con anterioridad en la familia (por ejemplo en trenes-own), aun así se prefirió trabajar con la rutina del modelo “GMU roads” a utilizar el primitivo de la extensión GIS.

```

to clear
  clear-all
  reset-ticks
end
to setup
  clear-all
  reset-ticks
  loadDataGis
  gis:set-world-envelope gis:envelope-of Alcaldias-dataset
  ::::::::::::::::::::::::::::::::::::::::::::Mostrar mapa de las alcaldías

  if show_Mapas? [
    showMapas
  ]
  ::::::::::::::::::::::::::::::::::::::::::::;Mostrar avenidas primarias y dibujar la red con agentes y colocar coches

  if show_Avenidas_Principales?[
    drawAP
  ]
  ::::::::::::::::::::::::::::::::::::::::::::;Mostrar avenidas secundarias y dibujar la red con agentes y colocar coches

  if show_Avenidas_Secundarias?[
    drawAS
  ]
  ::::::::::::::::::::::::::::::::::::::::::::;Mostrar línea 1 y dibujarla con agentes y colocar trenes

  if show_L1?[
    drawL1
    create-the-trenes-SL1
    create-the-trenes-NL1
    set promedio_ruta_L1_hosp []
    set promedio_ruta_L1_bom []
  ]

```

**Figura 3.20 Séptima sección del código: "to setup".
Fuente: Elaboración propia (2022).**

La figura 3.24 contiene las funciones para los botones “to clear” y “to setup” que aparecen en la figura 3.13, la función “to setup” está compuesta de otras funciones previamente establecidas para que fuera más fácil de leer el código así como detectar más fácilmente su debido funcionamiento, también se generan listas para cada línea si se desean habilitar que pertenecen a las variables globales, en este caso se observa que se generaron las listas “promedio_ruta_L1_hosp []” y “promedio_ruta_L1_bom []” las cuales sirven para almacenar las distancias de las rutas más cortas que se van generando para hospitales y estaciones de bomberos respectivamente y poder sacar el promedio.

```

..... Zoom L1
to zoom_L1
clear-all
reset-ticks
loadDataGIS

gis:set-world-envelope gis:envelope-of L1-estaciones-dataset

if show_Mapas? [
showMapas
]
if show_Avenidas_Secundarias?[
drawAS
]

;;;;;Dibujando red de avenidas principales y colocando coches
if show_Avenidas_Principales?[
drawAP
;create-the-cars-AP
]

if show_Bomberos?[
drawBomberos
]

if show_ISSSTE?[
drawISSSTE
]

if show_Centros_Salud?[
drawCenSal
]
drawL1
create-the-trenes-SL1
create-the-trenes-NL1
set promedio_ruta_L1_hosp []
set promedio_ruta_L1_bom []
end

```

**Figura 3.21 Octava sección del código: "Función Zoom".
Fuente: Elaboración propia (2022).**

La función “zoom_LX” donde x es el número o letra de cada línea del metro, es muy importante porque representa un sub-modelo donde es más fácil analizar el comportamiento del incidente para la línea en cuestión. Con el primitivo que se observa en la figura 3.25 “gis:set-world-envelope gis:envelope-of” seguido del GIS para cada línea, se restringe el mundo de la (figura 3.12) para que esté delimitado por las terminales de la línea en cuestión. Además dentro de esta función se encuentra el código que responde a los “switches” de la (figura 3.13) que habilitan o deshabilitan la visualización de los hospitales del ISSSTE los centros de salud, las estaciones de bomberos, el número de trenes que se desea para la línea y si se desea visualizar las avenidas principales y secundarias. Finalmente también se crean las listas para generar los promedios de las distancias de las rutas más cortas entre el lugar del incidente y el hospital o estación de bomberos más cercana para la línea en cuestión.

```

..... Funcion de movimiento de trenes
to movimiento_trenes
ask trenes!

ifelse (descompuesto? = true) [stop] [
  set prob_cc random-float 100 if prob_cc = 0 [ set prob_cc random-float 100]
  set prob_heridos random-float 100 if prob_heridos = 0 [ set prob_heridos random-float 100]
  set prob_incendio random-float 100 if prob_incendio = 0 [ set prob_incendio random-float 100 ]
  set mynode one-of (out-link-neighbors) of mynode
  ifelse (mynode = nobody) [renueva] [face mynode ifelse (prob_cc) = probabilidad_corto_circuito and (prob_cc) != 0 [
    set descompuesto? true
    construyentesTC
    set previouscolor color
    set color yellow
    clear-output
    output-type "Se descomposo tren" output-type " " output-type who output-type " " output-type "Con direccion " output-type direccion

    (ifelse (prob_heridos) = probabilidad_heridos and (prob_heridos) != 0 [set heridos? true crearDestin llamar_ambulancia if (prob_incendio) = probabilidad_incendio and (prob_incendio) != 0 [set incendio? true crearDestin llamar_bomberos]
    (prob_incendio) = probabilidad_incendio and (prob_incendio) != 0 [set incendio? true set heridos? false crearDestin llamar_bomberos][clear-output output-type "El tren ha sido reparado." set color previouscolor set descompuesto? false
    set prob_cc random-float 100 if prob_cc = 0 [ set prob_cc random-float 100]
    set prob_heridos random-float 100 if prob_heridos = 0 [ set prob_heridos random-float 100]
    set prob_incendio random-float 100 if prob_incendio = 0 [set prob_incendio random-float 100 ] ]][move-to mynode]]
  set previousnode mynode
  set mynode mynode
]
end

```

Figura 3.22 Novena sección del código: "Movimiento de trenes".
Fuente: Elaboración propia (2022).

En la figura 3.26 se expresa la función que gobierna el movimiento de trenes, sin duda es la función más relevante del modelo ya que contiene el incidente estudiado así como todos los escenarios posibles considerados (descritos previamente en el modelo conceptual) para entender cómo funciona véase la figura 3.6 que describe de manera simplificada pero certera la función.

```

to renueva
  if direccion = "Pantitlán L1" [hatch-trenes 1 [
    set color white
    set direccion "Observatorio L1"
    set shape "train passenger car"
    ;set hidden? true
    set size 2
    set prob_cc 100
    set pasajeros random 1000 + 100
    set mynode one-of verticesSL1-here ] die]
  if direccion = "Observatorio L1" [hatch-trenes 1 [
    set color red
    set direccion "Pantitlán L1"
    set shape "train passenger car"
    ;set hidden? true
    set size 2
    set prob_cc 100
    set pasajeros random 1000 + 100
    set mynode one-of verticesNL1-here ] die]
  if direccion = "Taxqueña L2" [hatch-trenes 1 [
    set color white
    set direccion "Cuatro Caminos L2"
    set shape "train passenger car"
    ;set hidden? true
    set size 2
    set prob_cc 100
    set pasajeros random 1000 + 100
    set mynode one-of verticesNL2-here ] die]

```

Figura 3.23 Décima sección del código: "Función 'renueva' para el cambio de dirección".
Fuente: Elaboración propia (2022).

En la décima sección del código se tiene parte de la función "renueva" que es parte del movimiento de los trenes y les permite cambiar de dirección sobre su misma línea sin tener conflictos con otros agentes. Aunque propiamente un mismo agente no realiza el movimiento sino que crea otro agente en la dirección opuesta y el agente padre muere.

```

to llamar_ambulancia
  set incidentesAmbulancias IncidentesAmbulancias + 1
  ifelse show_ISSSTE? or show_Centros_Salud? [
    clear-output
    output-type "Se revisó el tren, hay heridos, llamando ambulancia." output-type ""
    let prob_recursos_ISSSTE random 2
    let prob_recursos_CENSAL random 2
    ask verticesISSSTE [
      ifelse ISSSTE_Resources = "low" [set recursos_ISSSTE random 2] [set recursos_ISSSTE 2 set prob_recursos_ISSSTE 2]
      if prob_recursos_ISSSTE = recursos_ISSSTE [hatch-ambulancias 1 [
        set color white
        set shape "ambulance"
        set size 1.5
      ]]
    ]
    ask verticesCENSAL [
      ifelse Centros_Salud_Resources = "low" [set recursos_CENSAL random 2] [set recursos_CENSAL 2 set prob_recursos_CENSAL 2]
      if prob_recursos_CENSAL = recursos_CENSAL [hatch-ambulancias 1 [
        set color white
        set shape "ambulance"
        set size 1.5
      ]]
    ]
  ]
  ask turtle Destin [hatch-verticesAP 1 [set shape "face happy" set size 1 set hidden? true move-to min-one-of ambulancias [distance myself] set Origin who create-EnlaceAP-with min-one-of other verticesAP [distance myself]]]
]
[user-message "Está desactivada la visualización de Hospitales" stop]
if user_rutina_Dijkstra? = true [create-shortest-path]
let distancia_ruta 1
let ruta []
ask turtle Origin [foreach (n/paths-to turtle Destin) [x -> ask x [set distancia_ruta lput(link-length) distancia_ruta] set ruta n/paths-to turtle Destin]
print (word "La ruta más corta entre el hospital más cercano y el incidente de la línea " linea " es: " ruta " para una distancia total de: " sum distancia_ruta)
if linea = "1" [set promedio_ruta_l1_hosp lput (sum distancia_ruta) promedio_ruta_l1_hosp set distancia_media_hospital_l1 mean promedio_ruta_l1_hosp]
if linea = "2" [set promedio_ruta_l2_hosp lput (sum distancia_ruta) promedio_ruta_l2_hosp set distancia_media_hospital_l2 mean promedio_ruta_l2_hosp]
if linea = "3" [set promedio_ruta_l3_hosp lput (sum distancia_ruta) promedio_ruta_l3_hosp set distancia_media_hospital_l3 mean promedio_ruta_l3_hosp]
if linea = "4" [set promedio_ruta_l4_hosp lput (sum distancia_ruta) promedio_ruta_l4_hosp set distancia_media_hospital_l4 mean promedio_ruta_l4_hosp]
if linea = "5" [set promedio_ruta_l5_hosp lput (sum distancia_ruta) promedio_ruta_l5_hosp set distancia_media_hospital_l5 mean promedio_ruta_l5_hosp]
if linea = "6" [set promedio_ruta_l6_hosp lput (sum distancia_ruta) promedio_ruta_l6_hosp set distancia_media_hospital_l6 mean promedio_ruta_l6_hosp]
if linea = "7" [set promedio_ruta_l7_hosp lput (sum distancia_ruta) promedio_ruta_l7_hosp set distancia_media_hospital_l7 mean promedio_ruta_l7_hosp]
if linea = "8" [set promedio_ruta_l8_hosp lput (sum distancia_ruta) promedio_ruta_l8_hosp set distancia_media_hospital_l8 mean promedio_ruta_l8_hosp]
if linea = "9" [set promedio_ruta_l9_hosp lput (sum distancia_ruta) promedio_ruta_l9_hosp set distancia_media_hospital_l9 mean promedio_ruta_l9_hosp]
if linea = "A" [set promedio_ruta_lA_hosp lput (sum distancia_ruta) promedio_ruta_lA_hosp set distancia_media_hospital_lA mean promedio_ruta_lA_hosp]
if linea = "B" [set promedio_ruta_lB_hosp lput (sum distancia_ruta) promedio_ruta_lB_hosp set distancia_media_hospital_lB mean promedio_ruta_lB_hosp]
if linea = "12" [set promedio_ruta_l12_hosp lput (sum distancia_ruta) promedio_ruta_l12_hosp set distancia_media_hospital_l12 mean promedio_ruta_l12_hosp]
ifelse incendio? = true [set heridos? false ask turtle Origin [die]] [tren_reparado]
end

```

Figura 3.24 Onceava sección del código: “Función ‘llamar ambulancia’ para generar la ruta más corta”.

Fuente: Elaboración propia (2022).

En la figura 3.28 se describe otra función muy importante que es “llamar ambulancia” aunque se refiere concretamente a llamar a un hospital se le dejó ese nombre. En esta función se genera la ruta más corta desde el hospital al tren con el incidente. Esta función es activada o llamada cuando un tren ha tenido un incidente a su vez ha tenido usuarios heridos (esto debido a una probabilidad determinada por el usuario del modelo) al final de esta función se procede a llamar a la estación de bomberos en caso de que también se haya generado un incendio o en caso contrario se repara el tren.

3.9. OPTIMIZACIÓN EN EL MODELO DE SIMULACIÓN

En el modelo de simulación basado en agentes que se plantea en este trabajo, se introduce la optimización al calcular las rutas más cortas en la red de nodos y enlaces generada por las avenidas principales de la Ciudad de México (Secretaría de Movilidad. 28 enero 2021). La ruta más corta se calcula mediante el algoritmo de Dijkstra. Existe otro nivel de optimización introducido en este modelo. Se considera la ruta más corta desde el tren que sufre el incidente, al hospital o estación de bomberos más *cercana*, es decir, que se selecciona el hospital o estación de bomberos cuya distancia lineal sea mínima al lugar del incidente. Un tercer nivel en cuanto a esta optimización comienza cuando los recursos en los hospitales y estaciones de bomberos no son altos, puesto que en ese caso sólo algunos hospitales o estaciones de bomberos estarán disponibles y de entre ellos se seleccionará el más cercano y luego se calculará la ruta más corta. El orden en que se realizan las operaciones de optimización son:

1) Revisión de los recursos de los servicios de emergencia en el área definida por el contorno de las terminales de la línea del metro.

2) Localización del servicio de emergencia más cercano y que cuente con recursos para atender el incidente. La obtención de las coordenadas de los servicios de emergencia, se describe en la sección 3.4 de este trabajo de investigación.

3) Cálculo de la ruta más corta desde el servicio de emergencia, más cercano y con recursos, al lugar del incidente.

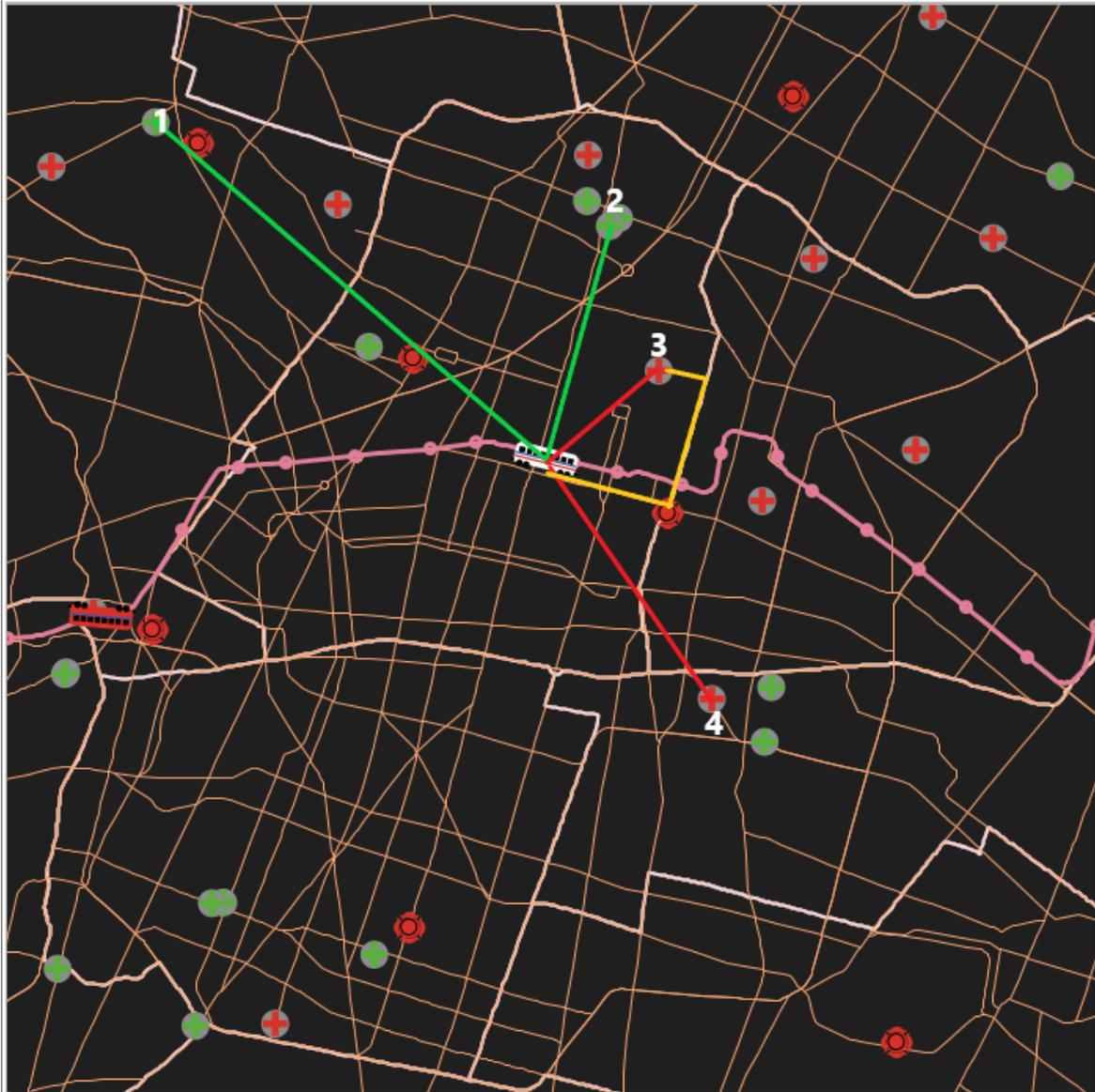


Figura 3.25 Ejemplo de optimización en el modelo de simulación basado en agentes.

Fuente: Elaboración propia (2022).

Como ejemplo, véase la figura 3.29, suponiendo que el tren blanco en el centro haya tenido un incidente y usuarios hayan resultado lesionados, entonces primero se revisan todos los hospitales del ISSSTE y Centros de Salud que tengan

Como ejemplo, en la figura 3.30 para la línea 1 del metro cuando los hospitales no tienen recursos siempre, en rojo se observa la ruta más corta (RMC), que recorre los enlaces: [5199 403 402 398 4230 133 134 2967 2968 1913 3511 5184] para una distancia total de 9.694457589384866 píxeles, cada uno de los números dentro de los corchetes representa la identificación de un agente, por lo tanto se forma un enlace entre el agente 5199 (el hospital) y el 403 (nodo de avenidas principales), el 403 y el 402 (nodo de avenidas principales) y así sucesivamente hasta llegar al 5184 (tren del incidente en color amarillo). La RMC se calculó en este ejemplo mediante el Algoritmo Dijkstra de (Gil Álvaro, 2012) así como por medio de la extensión "Networks" de NetLogo™. En la figura 3.30 se aprecian unas líneas grises claro que en la figura 3.29 no aparecen, estas líneas grises representan la red de avenidas principales simplificada debido a la cantidad de nodos en el GIS original, además, los hospitales que tienen recursos y por lo tanto están disponibles tienen una ambulancia de color blanco encima.

3.10. IMPLEMENTACIÓN DEL ALGORITMO DIJKSTRA EN NETLOGO™

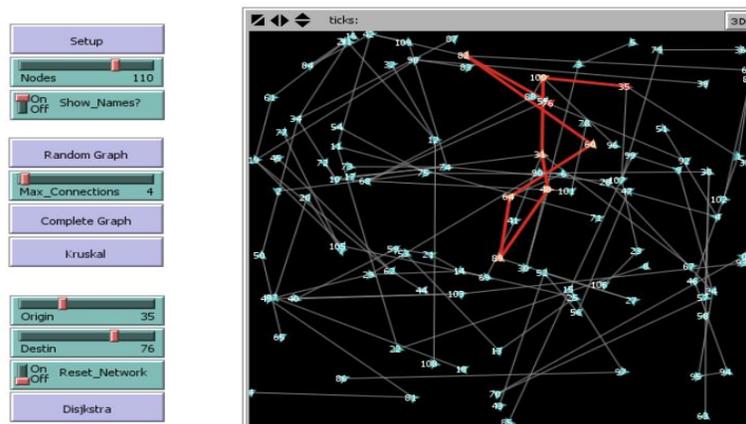


Figura 3.27 Algoritmo de Dijkstra en NetLogo™.
Fuente: (Gil Álvaro, 2012).

En NetLogo™, la extensión Networks (Nw) se puede usar para encontrar el camino más corto usando el primitivo "nw:path-to-target-turtle", que según Wilensky (1999), encuentra el camino más corto hacia la tortuga objetivo e informa la ruta real entre la tortuga de origen y la de destino. La ruta se informa como la lista de enlaces que constituyen la ruta. Si bien no se especifica qué algoritmo se usa en la primitiva nw:path-to de la extensión Networks para calcular el camino más corto, se comparó con una rutina desarrollada por Gil (2012) usando el Algoritmo de Dijkstra en NetLogo™.

Luego de un número considerable de pruebas con el código de (Gil Álvaro, 2012) a pesar de tener la ventaja de poder señalar visualmente con rojo la ruta y cambiar

el color de los demás agentes, la mayor desventaja que presentó es que al ser adaptado a ciertas líneas presentaba problemas de rendimiento (no lograba obtener la ruta más corta y dejaba de responder NetLogo™), en especial con las líneas 3, 4, 5, 7, A y B. Debido a esto se decidió trabajar con la extensión de “Networks” y se comparó con los resultados obtenidos con el código de (Gil Álvaro, 2012) que en cuanto a ruta de enlaces era la misma, sin embargo, el primitivo “nw:distance-to” no provee de una distancia lineal recorrida en la ruta más corta sino el número de enlaces que conforman la ruta, esto se resolvió escribiendo otras líneas de código, sin embargo, en la documentación oficial de NetLogo™ (Wilensky U, 1999) no se menciona mediante qué algoritmo se calcula la ruta más corta en el primitivo “nw:path-to”, sin embargo, debido a que los resultados coinciden con lo obtenido del código de (Gil Álvaro, 2012) se concluye que utiliza el algoritmo de Dijkstra.

3.11. VERIFICACIÓN DEL MODELO

La verificación del modelo fue realizada a través de la combinación de tres enfoques de modelación. El principio de diseño de ABMS, la de espectro completo y la modelación iterativa. A continuación se describe brevemente cada enfoque y al final se detalla cómo se conformaron los tres enfoques como proceso constante de verificación.

El primer enfoque fue el *Principio de Diseño ABMS* (ABMS Design Principle) (Wilensky and Rand, 2015, pp. 160-161) que consiste en empezar simple e ir añadiendo sólo las capas de complejidad necesarias para explicar el fenómeno realista. Hay dos componentes principales de este principio. El primer componente se refiere a comenzar con el conjunto más simple de agentes y reglas de comportamiento que se puedan usar para explorar el sistema que desea modelar. El segundo componente consiste en siempre tener en mente la pregunta o preguntas que guían la modelación, lo que significa no agregar nada al su modelo que no ayude a responder las preguntas. Este principio básico de diseño de ABM es útil de varias maneras. El *Principio de Diseño ABMS*, es útil ya que al examinar cada componente adicional para determinar si es necesario para responder a la pregunta de investigación que está buscando, se reduce la posibilidad de introducir ambigüedades, redundancias e inconsistencias en el modelo.

El segundo enfoque consideró el *Full Spectrum Modeling* o modelación de espectro completo, definida por (Wilensky and Rand, 2015, p. 354) como: “...modelar un fenómeno en múltiples niveles de detalle, es decir, no solo en el nivel de lo elaborado o simple, sino todos los niveles intermedios”. Este enfoque a veces puede tomar la forma de una serie de modelos, en los que uno comienza con modelos simples y se vuelven progresivamente más realistas, o bien se comienza con modelos realistas y se simplifican progresivamente. Al construir modelos en múltiples niveles diferentes de elaboración y realismo, es posible extraer principios generales del fenómeno que se analiza e incorporar esos principios generales en modelos simples. Entre las ventajas de utilizar este enfoque de modelación se encuentran: 1) Los investigadores pueden calibrar los

modelos elaborados y realistas con datos del mundo real para mostrar cómo esos modelos crean descripciones precisas del mundo que los rodea, 2) Los modelos simples permiten explorar en profundidad un pequeño número de mecanismos y mostrar que esos mecanismos pueden tener efectos profundos en nuestros resultados.

El tercer enfoque utilizado en la creación del modelo fue el “*Iterative Modeling*” o modelación iterativa la cual consiste en armar un modelo in pequeñas fases y tener retroalimentación frecuente entre el diseño del modelo conceptual y el proceso de implementación del modelo. De acuerdo con (Wilensky & Rand, 2015, p.355) La metodología de modelado iterativo postula que es útil obtener retroalimentación a lo largo de los procesos de diseño e implementación del modelo. Esto le permite al diseñador del modelo ver el modelo en varias etapas de finalización y realizar cambios en el modelo según sea necesario. Así mismo establecen las siguientes etapas en el modelado iterativo: (1) diseñar el modelo con base en datos/teorías iniciales, (2) implementar el modelo con mecanismos básicos, (3) revisar los datos/teorías con base en la realización de sub-especificaciones o resultados de ejecuciones del modelo, (4) revisar la implementación con base en los datos/teorías, y (5) repetir los pasos 1 – 4.

Esta combinación de los enfoques de diseño de modelos podrían considerarse un diseño tipo (bottom-up). En resumen, la verificación se realizó como un proceso desde el inicio de la implementación del modelo el cual se puede describir como un modelo grande de la red del metro y las vialidades primarias y secundarias de la CDMX, con muchos interruptores (switches) que permiten encender y apagar diferentes mecanismos (esto implica la modelación de espectro completo), estos mecanismos son varios modelos más pequeños y simples (Principio de diseño ABMS) de cada línea del metro durante todo el proceso de modelado se verificó constantemente con y en comparación con el modelo conceptual, evolucionando con él (Modelado iterativo).

3.12. VALIDACIÓN DEL MODELO

La validación del modelo fue realizada mediante lo que plantean, Rand & Rust (2011), donde dicen que hay dos ejes a lo largo de los cuales se consideran los problemas de validación. De acuerdo a lo que proponen Rand & Rust (2011) se desarrollaron los espectros de las figuras 3.28 y 3.29, el desarrollo de estos espectros surge a partir de que el autor de este trabajo de investigación entendió que la validez de un modelo es relativa a las preguntas para las que se utiliza, entonces la validación de un modelo puede considerarse más como un espectro dinámico, es decir, un modelo nunca es inherentemente válido; más bien, su validez proviene del contexto para el que se utiliza.

De la figura 3.28 y 3.29, el primer eje de validación es el nivel en el que ocurre el proceso: microvalidación (partes importantes del comportamiento individual del agente) y macrovalidación (partes importantes del sistema). El segundo eje es el nivel de detalle del proceso de validación: “*face validation*” o validación de

apariencia (las propiedades y los mecanismos del modelo se parecen al mundo real) y “*empirical validation*” o validación empírica (el modelo genera datos que corresponden a patrones similares de datos del mundo real). Encontrar los parámetros y las condiciones iniciales que hacen que el modelo coincida con los datos del mundo real es un proceso auxiliar en la validación empírica (*empirical validation*) llamado calibración. “La calibración implica el uso de datos empíricos para ajustar los parámetros y mecanismos de un modelo de modo que pueda usarse para examinar una situación particular” (Wilensky & Rand, 2015, p. 332).

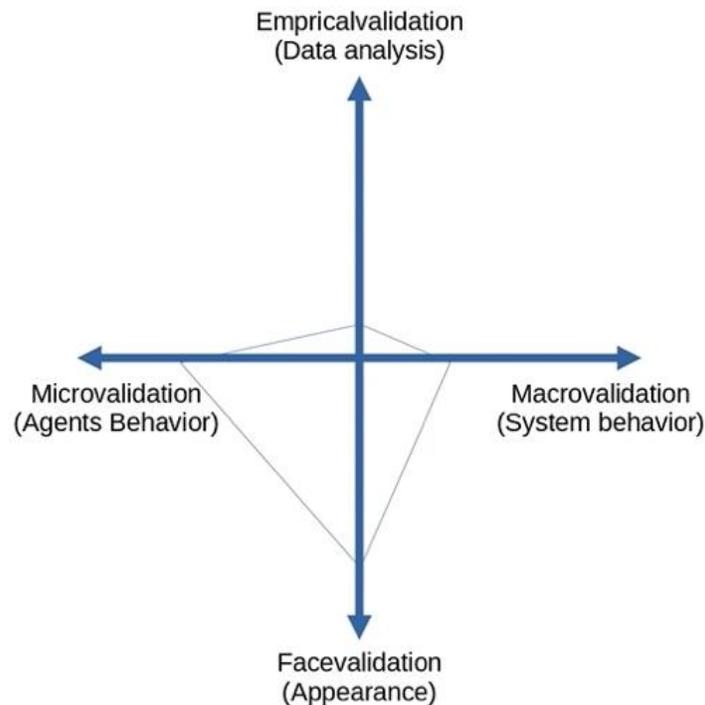


Figura 3.28 Primer espectro de validación.
Fuente: Elaboración propia (2022).

El espectro de validación de la figura 3.28, fue el primer espectro de validación del modelo de simulación optimización basado en agentes presentado en este estudio y se realizó de manera previa a la implementación de los algoritmos de ruta más corta. Representa entonces una validación en cuanto a: 1) el movimiento de los trenes y las propiedades de éstos (comportamiento de los agentes), 2) los mapas sean reproducidos de manera correcta y fiel a lo establecido en los sistemas de información geográfica (apariencia). La primera validación demuestra que el modelo de simulación-optimización basado en agentes, tiene una alta validación facial (ver figura 3.26) y microvalidación (figura 3.20).

Una vez implementado el algoritmo de Dijkstra y la extensión de “Networks” de NetLogo™ se realizó una segunda validación, especialmente para aumentar la validez en cuanto al análisis de datos empíricos y la macrovalidación, véase figura 3.29.

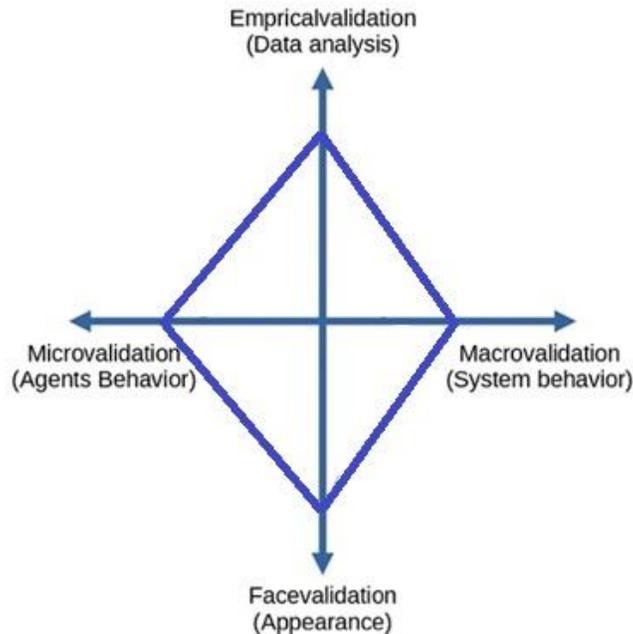


Figura 3.29 Segundo espectro de validación.
Fuente: Elaboración propia (2022).

Entonces de la figura 3.29, la validación empírica aumenta ya que el modelo de simulación-optimización de este trabajo de investigación produce resultados a partir de la implementación del algoritmo Dijkstra y la extensión *Networks*.

Cabe recordar que los modelos son simplificaciones de la realidad, por lo que no se puede lograr una validación perfecta. Además, la validación es relativa a las preguntas que se tienen en mente al momento de realizar este proceso y dinámica ya que pueden surgir más preguntas con el tiempo. A continuación se escriben las preguntas que se utilizaron para la validación del modelo de simulación-optimización de este trabajo de investigación que dieron lugar a los espectros de las figuras 3.28 y 3.29.

1) Microvalidación:

- a. ¿Los agentes de tipo tren se mueven a lo largo de las líneas de la red del STC?
- b. ¿Los agentes de tipo tren tienen usuarios?
- c. ¿Los agentes de tipo tren respetan el movimiento?
- d. ¿Los agentes tipo hospitales, centros de salud y estaciones de bomberos cuentan con propiedades que les asignen recursos?

2) Macrovalidación

- a. ¿Se representa el movimiento adecuado de los trenes en la red?
- b. ¿Se representa el incidente *cortocircuito por llanta pinchada*?
- c. ¿Se observa la interacción entre los hospitales públicos, centros de salud y estaciones de bomberos cuando sucede el incidente *cortocircuito por llanta pinchada*?

- 3) Validación de apariencia:
 - a. ¿Se representa de manera correcta la red del STC metro?
 - b. ¿Se representa de manera correcta el mapa de la CDMX?
 - c. ¿Se representan de manera correcta las avenidas principales de la CDMX?
 - d. ¿Se representan de manera correcta las ubicaciones de hospitales públicos, centros de salud y estaciones de bomberos?
- 4) Validación empírica:
 - a. ¿El modelo genera datos a partir de la ejecución del algoritmo Dijkstra?
 - b. ¿El modelo genera datos a partir de la ejecución de la extensión *Networks*?
 - c. ¿El modelo genera datos a partir de la ejecución sucesiva del algoritmo Dijkstra y la extensión *Networks*?

3.13. REPLICACIÓN O REPRODUCCIÓN DEL MODELO

De acuerdo con (Wilensky & Rand, 2015, p. 338): “Un modelo original y un modelo replicado asociado pueden diferir en al menos seis dimensiones: (1) tiempo, (2) hardware, (3) idiomas, (4) juegos de herramientas, (5) algoritmos y (6) autores.”

(Wilensky & Rand, 2015, pp. 338 – 340) continúan mencionando que una replicación exitosa es aquella en la que los replicadores pueden establecer que el modelo replicado crea resultados suficientemente similares a los del original. Esto no significa necesariamente que los dos modelos tengan que generar exactamente los mismos resultados, además, la importancia de una replicación exitosa de un experimento físico radica en que hace avanzar el conocimiento científico porque demuestra que el diseño y los resultados del experimento pueden generarse repetidamente para mostrar que los resultados originales no fueron anomalías excepcionales o estadísticas, de esta forma la replicación respalda el proceso de verificación del modelo porque si dos implementaciones distintas de un modelo conceptual son capaces de producir los mismos resultados, aumenta la confianza en que el modelo implementado captura con precisión el modelo conceptual y a su vez la replicación promueve la validación del modelo porque la validación es un proceso que determina una correspondencia entre los resultados de un modelo implementado y las medidas del mundo real.

Finalmente (Wilensky & Rand, 2015, pp. 339 – 340) establecen tres categorías de estándares de replicación: 1) La primera categoría de estándares de replicación, "identidad numérica", implica mostrar que los modelos original y replicado producen exactamente los mismos resultados numéricos. 2) La segunda categoría de estándares de replicación es la “equivalencia distribucional”. Cuyo objetivo es mostrar que los dos modelos implementados son suficientemente similares entre sí según las medidas estadísticas. 3) La última categoría de estándares de

replicación es “alineación relacional.” Existe alineación relacional si los resultados de los dos modelos implementados muestran relaciones cualitativamente similares entre las variables de entrada y salida.

Con base a lo descrito previamente y para facilitar el proceso de replicación así como para proveer del carácter científico al modelo de simulación optimización que se plantea, se decide compartir los archivos GIS y el archivo de NetLogo™ en la siguiente liga:

<https://github.com/AdrianGutierrezGomez/ModeloTesis.git>

CAPÍTULO IV: EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

4. DISEÑO DE EXPERIMENTOS

4.1. CARACTERÍSTICAS DE LA COMPUTADORA UTILIZADA

A continuación, en la (figura 4.1) se especifican las características de la computadora utilizada para realizar los experimentos.

[Ver información básica acerca del equipo](#)

Edición de Windows

Windows 10 Pro
© Microsoft Corporation. Todos los derechos reservados.

Sistema

Procesador:	Intel(R) Pentium(R) CPU G2030 @ 3.00GHz 3.00 GHz
Memoria instalada (RAM):	8.00 GB (7.89 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo:	Usuario-PC
Nombre completo de equipo:	Usuario-PC
Descripción del equipo:	
Grupo de trabajo:	WORKGROUP

Activación de Windows

Windows está activado [Lea los Términos de licencia del software de Microsoft](#)

**Figura 4.1 Características del equipo utilizado para realizar los experimentos.
Fuente: Elaboración propia (2022).**

4.2. BEHAVIORSPEACE

El diseño y ejecución de los experimentos se realiza con la herramienta BehaviorSpace incluida en NetLogo™. Lo que esta herramienta realiza es la ejecución un modelo una gran cantidad de veces, variando sistemáticamente la configuración del modelo y registrando los resultados de cada ejecución del modelo. Este proceso a veces se denomina "barrido de parámetros", lo que permite generar y explorar el "espacio" del modelo de posibles comportamientos y determinar qué combinaciones de configuraciones provocan los comportamientos de interés.

En el diseño de experimentos se consideran los escenarios planteados originalmente en el modelo conceptual:

- I. Cuando todos los hospitales y estaciones de bomberos cuentan con recursos.
 - A. Sólo sucede el incidente de cortocircuito sin heridos ni incendio.
 - B. Sucede el incidente de cortocircuito con heridos pero sin incendio.
 - C. Sucede el incidente de cortocircuito sin heridos pero con incendio.
 - D. Sucede el incidente de cortocircuito con heridos y con incendio.
- II. Cuando algunos (50% de probabilidad) hospitales y estaciones de bomberos cuentan con recursos.
 - A. Sólo sucede el incidente de cortocircuito sin heridos ni incendio.
 - B. Sucede el incidente de cortocircuito con heridos pero sin incendio.
 - C. Sucede el incidente de cortocircuito sin heridos pero con incendio.
 - D. Sucede el incidente de cortocircuito con herido y con incendio.

Tomando como ejemplo el escenario I-A en la línea 1 del metro, la interfaz de BehaviorSpace en NetLogo™ se visualiza en la (figura 4.2) y se describen los parámetros a variar.

- ["Projection" "Lambert_Conformal_Conic"] – para escoger la proyección
- ["show_Mapas?" true] – habilitar los mapas de la CDMX (alcaldías)
- ["show_Avenidas_Principales?" true] – habilitar las avenidas principales
- ["number-of-trenes" 8] – colocar ocho trenes sobre la línea 1
- ["usar_rutina_Dijkstra?" false] – es opcional usar la rutina de (Gil Álvaro, 2012) por lo tanto se deshabilita
- ["probabilidad_corto_circuito" 1] – se establece la probabilidad de cortocircuito como 1%
- ["probabilidad_heridos" 0] – probabilidad del 0% porque es el escenario I-A
- ["probabilidad_incendio" 0] – probabilidad del 0% porque es el escenario I-A
- ["show_Bomberos?" true] – habilitar visualización de estaciones bomberos
- ["show_ISSSTE?" true] – habilitar visualización de hospitales ISSSTE
- ["show_Centros_Salud?" true] – habilitar visualización de Centros de Salud
- ["ISSSTE_Resources" "high"] – siempre tiene recursos, escenario I
- ["Centros_Salud_Resources" "high"] – siempre tiene recursos, escenario I

- ["Fire_Station_Resources" "high"] – siempre tiene recursos, escenario I

Experiment

Nombre del Experimento

Varias las variables de la siguiente manera (atención a los paréntesis y comillas):

```
[{"Projection" "Lambert_Conformal_Conic"}
{"show_Mapas?" true}
{"show_Avenidas_Principales?" true}
{"number-of-trenes" 8}
{"usar_rutina_Dijkstra?" false}]
```

Indicar los valores a utilizar, p.e.:
["Mi-deslizador" 1 2 7 8]
o bien, especificar inicio, incremento, y valor final, p.e.:
["Mi-deslizador" [0 1 10]] (atención a los paréntesis adicionales)
para pasar de 0 a 10, con incrementos de 1.
También se puede variar max-pxcor, min-pxcor, max-pycor, min-pycor y random-seed.

Repeticiones
ejecutar cada combinación tantas veces como

Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Evaluar las ejecuciones utilizando estos indicadores:

```
IncidentesL1
IncidentesAmbulancias
IncidentesBomberos
distancia_media_hospital_L1
distancia_media_bom_L1
```

un indicador por línea; no se pueden
dividir indicadores en varias líneas

Evaluar las ejecuciones a cada paso
si no se marca, se evalúan ejecuciones al finalizar cada una.

Instrucciones de Configuración inicial:

Instrucciones de Ejecución:

Condición de fin de ejecución:

Instrucciones post-ejecución:

la ejecución se detiene cuando se cumple esta condición se ejecutan al finalizar cada ejecución

Límite de tiempo
se detiene la ejecución al alcanzar este número de pasos (0 = sin límite)

Figura 4.2 Ejemplo de experimento para el escenario I-A en la línea 1 del metro.
Fuente: Elaboración propia (2022).

Además se solicita que se cuenten los incidentes con la variable "IncidentesTotales", los incidentes relacionados con ambulancias que para este caso debería de permanecer en cero, sucede también en los incidentes

relacionados con bomberos, así mismo, se registra el promedio de las distancias de las rutas más cortas (desde el hospital más cercano al tren con el incidente) que se han generado a lo largo del experimento que también en este caso es cero ya que sólo se generan los incidentes sin necesidad de llamar a servicios de emergencia finalmente se repite diez veces el experimento y la condición de fin de ejecución del modelo es que se generen mil incidentes.

Para los escenarios I-B, I-C, I-D, II-B, II-C y II-D las probabilidades de heridos e incendio se establecieron como 50%. Finalmente como se generan ocho escenarios para cada línea, en total son 96 y cada escenario se ejecuta 5 veces lo que da un total de 480 ejecuciones del modelo, sin embargo, se espera que los datos de los escenarios I-A y II-A sean similares debido a que sólo se generan incidentes sin calcular rutas más cortas ni que estas sean afectadas por los recursos de los servicios de emergencia, además para la línea A se considera un caso especial exento de los experimentos debido a las propiedades que más tarde se describirán, por lo tanto en total quedan 88 escenarios.

**Tabla 4.1 Ejemplo de tabla resultado para el experimento línea 1 I-B.
Fuente: Elaboración propia (2022).**

BehaviorSpace results (NetLogo 6.2.2)					
Modelo_Tesis.nlogo					
Experimento_L1_IB					
08/30/2022 16:46:22:179 -0500					
min-pxcor	max-pxcor	min-pycor	max-pycor		
-15	15	-15	15		
[run number]	1	1	1	1	1
Projection	Lambert_Conformal_Conic				
show_Mapas?	true				
show_Avenidas_Principales	true				
number-of-trenes	8				
usar_rutina_Dijkstra?	false				
probabilidad_corto_circuito	1				
probabilidad_heridos	50				
probabilidad_incendio	0				
show_Bomberos?	true				
show_ISSSTE?	true				
show_Centros_Salud?	true				
ISSSTE_Resources	high				
Centros_Salud_Resources	high				
Fire_Station_Resources	high	Incidentes	Incidentes	distancia_media	distancia_media
[reporter]	IncidentesL1	Ambulancia	Bomberos	_hospital_L1	_bom_L1
[final]	1000	494	0	6.674391467	0
[min]	0	0	0	0	0
[max]	1000	494	0	8.551345753	0
[mean]	498.7171543	246.345125	0	6.713666354	0
[steps]	12748	12748	12748	12748	12748
[all run data]	IncidentesL1	IncidentesA	IncidentesB	distancia_media	distancia_media
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

4.3. CASOS ESPECIALES LÍNEA A y B

Tanto la línea A como la línea B del metro presentan características diferentes a las demás líneas, entre ellas es que aproximadamente la mitad de sus estaciones se encuentran en la Ciudad de México y la otra mitad se encuentran en el Estado de México, esto plantea ciertos problemas por cómo fue desarrollado el modelo. En primer lugar desde el modelo conceptual se estableció que se utilizarían las avenidas principales de la Ciudad de México, esto representa una desventaja tanto como para la línea A y B debido a que aproximadamente la mitad de la líneas no tendrían un nodo de las avenidas principales cercano, esta desventaja se ve aún más acentuada en la línea A debido a la poca disponibilidad de avenidas principales, figura 4.3, esto se reflejó directamente en el modelo porque en ocasiones sí se lograba obtener la ruta más corta pero en otras ocasiones no lograba debido a una mala conectividad con las avenidas principales.



Figura 4.3 Línea A sin visualización de alcaldías pero con visualización de avenidas principales.

Fuente: Elaboración propia (2022).

Las avenidas principales de la figura 4.3, no están conectadas en su totalidad porque así fue generado el SIG por parte de la Secretaría de Movilidad (28 enero 2021). En la figura 4.3, los trenes de color blanco se dirigen a la terminal

“Pantitlán” mientras que los de color rojo se dirigen a terminal “La Paz”. Otro problema son los pocos servicios de emergencia para la línea A (que sólo se visualizan los del lado de la Ciudad de México) como se muestra en la (figura 4.3) sólo se cuenta con una estación de bomberos, en los casos IIB, IIC y IID existía la posibilidad de que esa estación no contara con recursos, lo cual representaba un problema a la hora de ejecutar el modelo.

La línea B (figura 4.4) cuenta con más hospitales pero sólo una estación de bomberos más por lo que se modificó el código para que al menos una de las dos estaciones de bomberos tuviera recursos siempre para los escenarios descritos previamente, además no representaba un problema para el modelo debido a que tiene una mayor conectividad de avenidas principales a diferencia de la línea A. Son estas razones por las que se decidió excluir a la línea A de los experimentos.

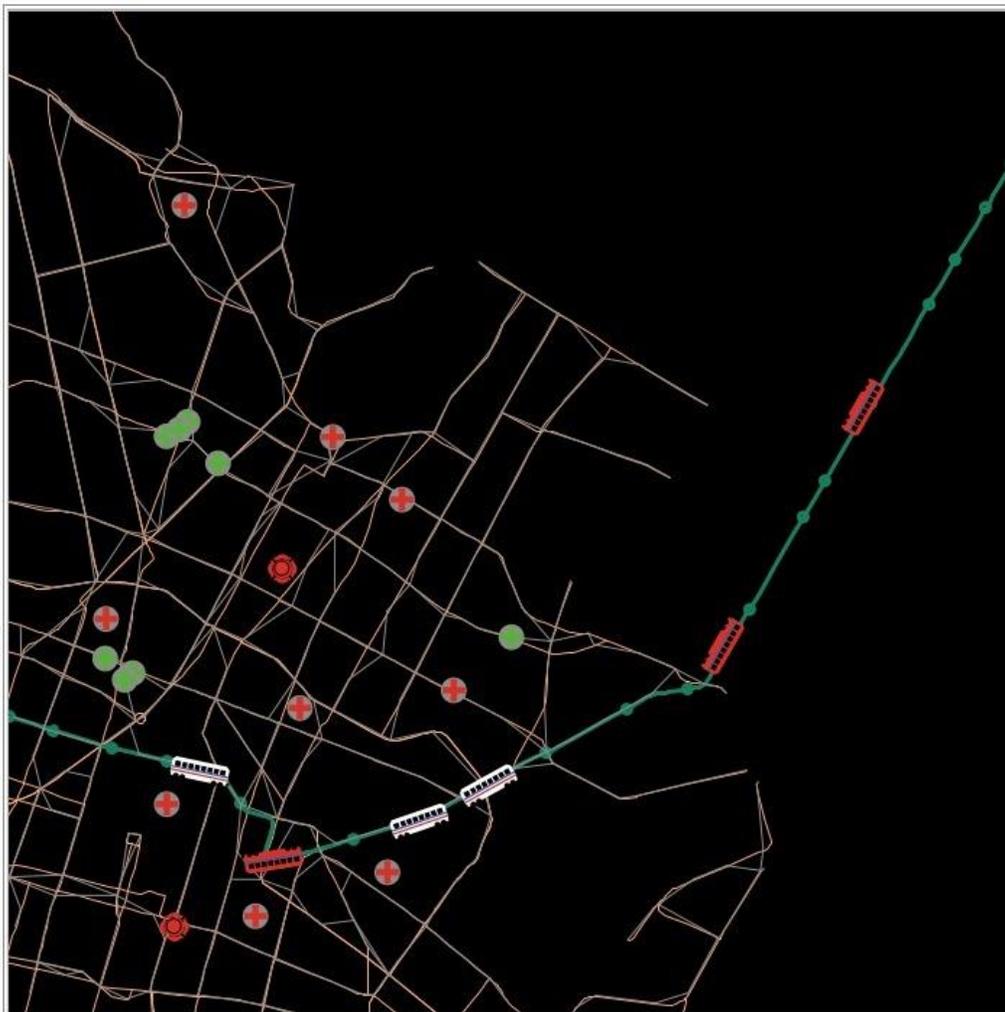


Figura 4.4 Línea B sin visualización de alcaldías pero con visualización de avenidas principales.

Fuente: Elaboración propia (2022).

Para un debido tratamiento de las líneas A y B, ejercicio que se deja para el futuro y para quien quiera seguir con esta investigación, se tendrían que considerar avenidas secundarias (sobre todo para el caso de la línea A), los servicios de emergencia que se encuentren en el Estado de México (tanto hospitales como estaciones de bomberos) y las avenidas principales del Estado de México pero sólo de los municipios que involucrados con las líneas A y B.

4.4. RESULTADOS DE LOS EXPERIMENTOS

Se hizo la recopilación de los datos de los 88 escenarios ejecutados por el modelo de simulación-optimización basado en agentes. A continuación se presentan tres gráficas que reúnen la información recopilada.

La gráfica de la figura 4.5 muestra el tiempo promedio, en tics, de las cinco ejecuciones para cada escenario de cada línea.

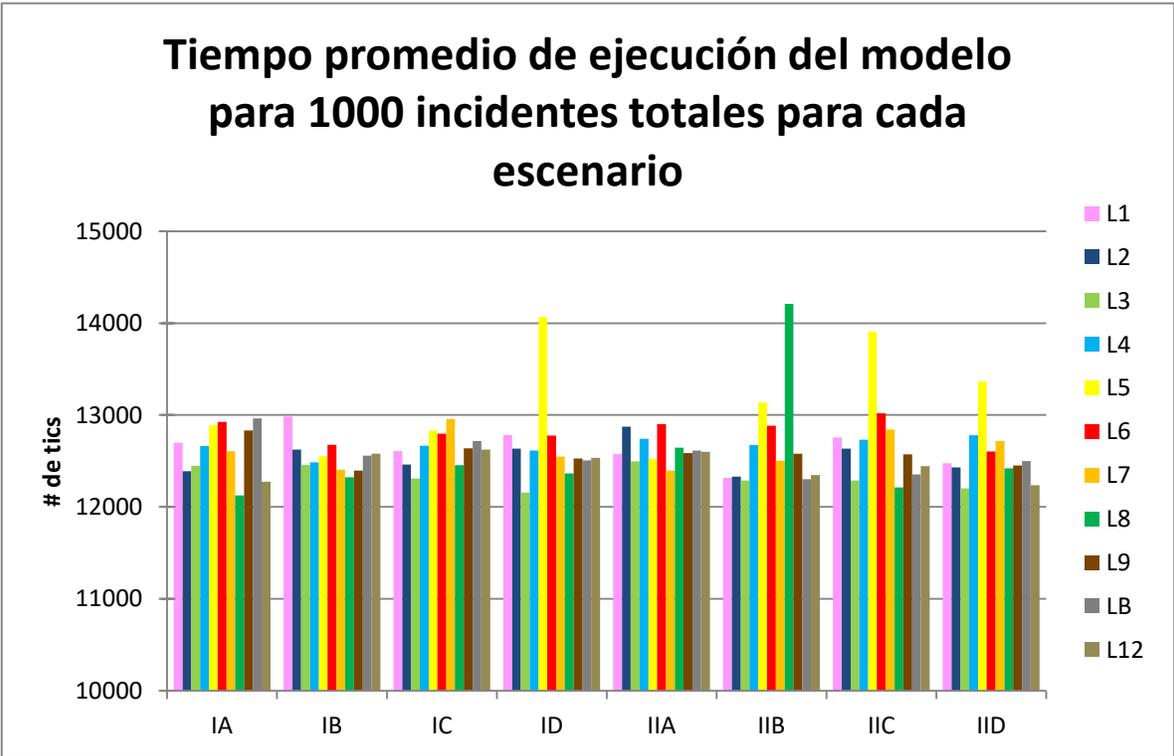


Figura 4.5 Tiempo promedio de ejecución del modelo para 1000 incidentes para cada escenario.

Fuente: Elaboración propia (2022).

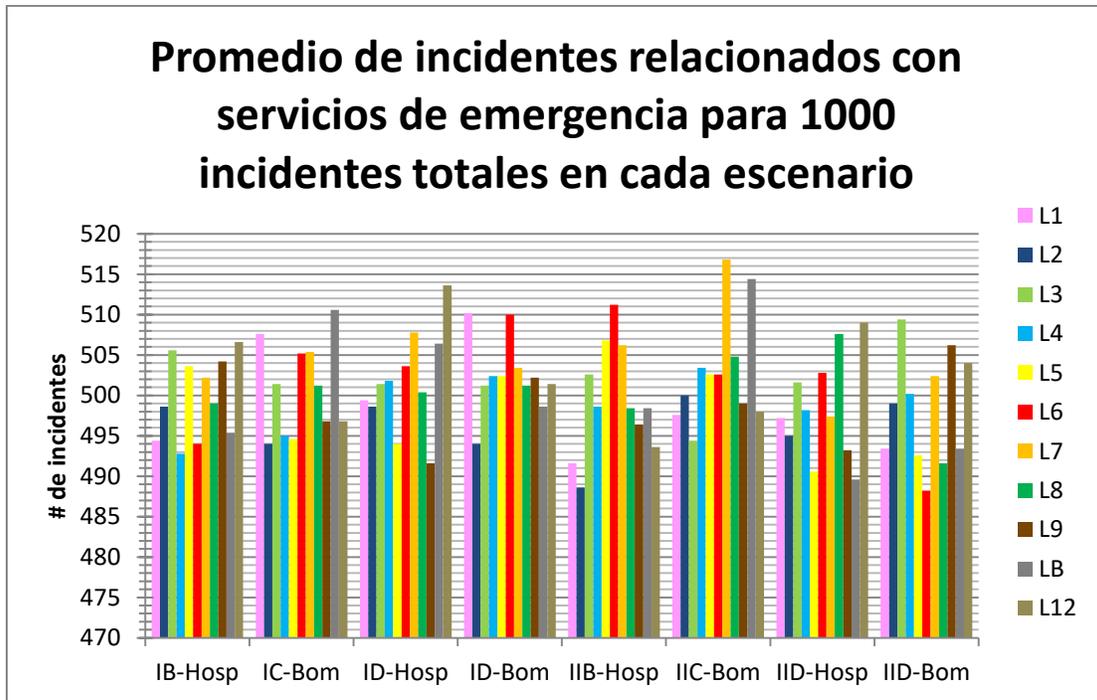


Figura 4.6 Promedio de incidentes relacionados con servicios de emergencia para 1000 incidentes totales en cada escenario.
Fuente: Elaboración propia (2022)

La gráfica presentada en la figura 4.6 muestra el promedio de incidentes relacionados con servicios de emergencia que surgieron a lo largo de 1000 incidentes, esta gráfica está relacionada a la anterior porque la primera (figura 4.5) habla del tiempo promedio (número de tics) que tardó el modelo en llegar a 1000 incidentes totales. La figura 4.6 muestra el promedio de incidentes relacionados a servicios de emergencia a lo largo de esos 1000 incidentes totales.

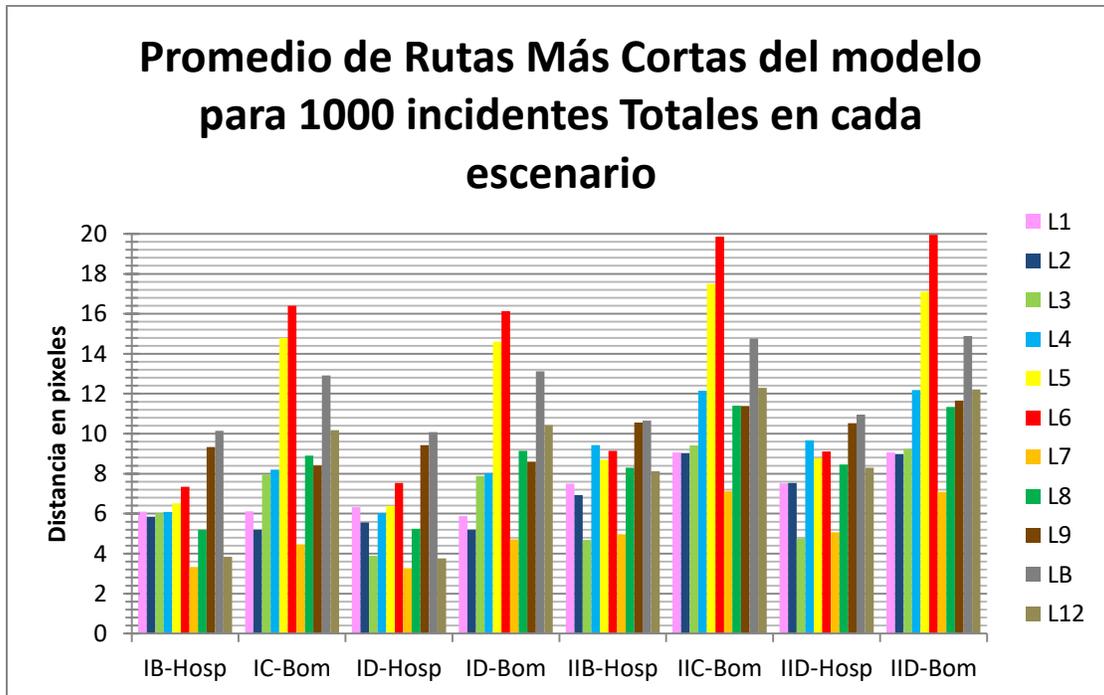


Figura 4.7 Promedio de Rutas Más Cortas del modelo para 1000 incidentes totales en cada escenario.

Fuente: Elaboración propia (2022).

Finalmente, la gráfica observada en la figura 4.7 presenta la media del promedio de rutas más cortas desde el servicio de emergencia más cercano al tren del incidente, sobre cualquier punto de una línea del metro

4.5. ANALISIS DE LOS RESULTADOS DE LOS EXPERIMENTOS

De la figura 4.5 se puede observar que el tiempo promedio de las ejecuciones del modelo de simulación-optimización se mantuvo dentro de un rango de 12,000 a 13,000 tics, para algunos escenarios de la línea 5 y el escenario IIB de la línea 8, esto se puede interpretar que en general el modelo es consistente en el rango de 12,000 a 13,000 tics en ejecutar 1000 incidentes, la variación que existe es en parte por la probabilidad del 1% del incidente y además de por las características de la computadora, en tiempo real el tiempo de cada conjunto de cinco ejecuciones del modelo para cada escenario estuvo entre un minuto y tres minutos, sin embargo, luego de correr los ocho escenarios de una línea la computadora parecía calentarse un poco por lo que no se continuaba con la ejecución.

La figura 4.6 es una comprobación de la aleatoriedad de los incidentes, no hay una clara repetición en los datos como era de esperarse, debido a que se tenía configurado un 50% de probabilidad de heridos y de incendio, por lo tanto, la mitad

de los 1000 incidentes totales debía de estar relacionada con alguno de los servicios de emergencia.

Finalmente, la gráfica de la figura 4.7 muestra el promedio del promedio de rutas más cortas desde el servicio de emergencia más cercano al lugar del incidente sobre cualquier punto de la línea correspondiente del metro. Este promedio de rutas se puede considerar como un indicador de la cercanía de los servicios de emergencia, ya que gracias a la simulación se produjeron 1000 incidentes en cualquier punto sobre una línea del metro; cada incidente tenía su ruta más corta desde el servicio de emergencia más cercano (origen) al incidente (destino) y la distancia de esta ruta se fue guardando a lo largo de la ejecución del modelo para obtener el promedio el cual se iba actualizando con cada incidente nuevo. En otras palabras, lo que pretende responder este indicador es tener un servicio de emergencia virtual para el promedio de la línea a partir de la pregunta: si sucede un incidente en cualquier punto de la línea ¿cuál es la distancia de la ruta más corta al servicio más cercano? Entonces un número cercano a cero se puede interpretar que dado un incidente en cualquier punto de la línea del metro no se requiere viajar mucho para encontrar un servicio de emergencia, en el caso contrario entre más alejado del cero esté el promedio entonces se tendrá que recorrer mayor distancia. Este indicador es más interesante cuando se consideran los casos II-B, II-C y II-D que introducen el concepto de disponibilidad de recursos, ya que no siempre se tendrán recursos los hospitales y estaciones de bomberos más cercanos podrían no estar disponibles entonces ¿cuál sería distancia a recorrer?

Dicho lo anterior se pueden sacar conclusiones sobre la disponibilidad de los servicios de emergencia con respecto a las líneas del metro, por ejemplo la línea 7 del metro en los escenarios I-B I-C y I-D a comparación de las demás, es la que en promedio tiene la ruta más corta a los servicios de emergencia esto puede deberse a la ubicación clave de hospitales y estaciones de bomberos cercanos a la línea, sin embargo, un observación interesante es que cuando se considera la disponibilidad de recursos en los escenarios II-B, II-C y II-D el incremento en la distancia es el doble para los hospitales y estaciones de bomberos, sin embargo, aun considerando este incremento en general la línea 7 tiene las distancias más cortas considerando todas las líneas. En contraste directo se encuentra la línea 6 del metro, que tiene las distancias más largas con respecto a las estaciones de bomberos lo que implica que si sucediera un incendio en algún tren tardarían más en llegar los bomberos, esto sugiere que se requieren más estaciones de bomberos próximas a la línea 6. La línea B presenta resultados altos aunque es de tener cierta cautela debido a sus características explicadas en la sección 4.3 de este capítulo.

En conclusión los escenarios I-B I-C y I-D, proporcionan un indicador sobre la cercanía de los servicios de emergencia en cuanto a rutas más cortas, lo que se traduce en atención a emergencias como un posible desarrollo de estrategias y planes para atender las líneas del metro con grandes distancias en rutas más cortas, mientras que los escenarios II-B, II-C y II-D son indicador sobre la resiliencia de los sistemas de servicios de emergencia relacionados al STC ya que

un cambio drástico entre las distancias de los escenarios I y II implicaría que en caso de baja disponibilidad los sistemas de servicios emergencia no son suficientes lo que implicaría que se necesitan más para aumentar esa resiliencia.

4.6. RESULTADOS GENERALES

En este estudio, se logra integrar modelos de simulación basados en agentes (ABMS), autómatas celulares (CA), el algoritmo de Dijkstra para resolver problemas complejos de optimización de rutas como la ruta óptima desde los servicios de emergencia hacía los lugares de incidente en las líneas del metro de la Ciudad de México utilizando el software NetLogo™.

Los resultados son los siguientes:

- La integración ABMS y GIS fue exitosa en este estudio. Ya que ha representado grandes desafíos debido a que el software NetLogo™ utiliza “*patches*” o parches como referencia espacial a los agentes. Sin embargo, para este estudio los parches no eran una representación adecuada de los mapas de GIS. Por lo tanto, fue necesario implementar una red basada en agentes para representar la red de metro y las avenidas primarias y secundarias.
- El movimiento de trenes se logró considerando a los trenes como agentes y su movimiento en una red basada en agentes.
- Incidentes en trenes. Se implementó como un evento pseudoaleatorio. Depende de la probabilidad establecida por el usuario y la probabilidad de cada tren durante cada movimiento, que es independiente de eventos pasados. También se representan incidentes complementarios como riesgo de incendio y usuarios heridos en el interior de los trenes. El tiempo de duración de los incidentes es variable y se clasifica en ocho escenarios dependiendo de los recursos:
 - Cuando todos los hospitales y estaciones de cuentan con recursos
 - Sólo sucede el incidente de cortocircuito sin heridos ni incendio.
 - Sucede el incidente de cortocircuito con heridos pero sin incendio.
 - Sucede el incidente de cortocircuito sin heridos pero con incendio.
 - Sucede el incidente de cortocircuito con heridos y con incendio.
 - Cuando algunos (50% de probabilidad) hospitales y estaciones de bomberos cuentan con recursos.
 - Sólo sucede el incidente de cortocircuito sin heridos ni incendio.

- Sucede el incidente de cortocircuito con heridos pero sin incendio.
 - Sucede el incidente de cortocircuito sin heridos pero con incendio.
 - Sucede el incidente de cortocircuito con herido y con incendio.
- Implementación del algoritmo Dijkstra y la extensión “*Networks*” para la obtención de las rutas más cortas.
- Desarrollo de posible indicador para evaluar la necesidad de servicios de emergencia así como la resiliencia de los servicios con respecto al STC.

CONCLUSIONES

Se diseñó, implementó y validó de manera exitosa, un sistema de simulación-optimización basado en agentes con el fin brindar servicios de emergencia desde hospitales públicos y estaciones de bomberos hacia el lugar donde se presenta el incidente denominado *cortocircuito por llanta ponchada* en la red del metro, mediante la ruta más corta establecida en avenidas principales de la Ciudad de México, utilizando el software NetLogo™.

Son de suma importancia los datos del sistema de información geográfica (SIG) para el modelo en NetLogo™, si los SIG tienen errores o son demasiado masivos, se pueden hacer correcciones y modificaciones en NetLogo™, sin embargo, debido a que no es su principal función, esto resulta demasiado complicado y laborioso, por lo tanto es recomendable que el tratamiento de a los datos de los SIG sea previo a cargarlos en NetLogo™, tanto para facilitar el proceso de modelación como su avance. En este trabajo hubo problemas con los datos de SIG que se resolvieron tanto de manera previa como a la hora de cargarlos en NetLogo™ y es parte de la razón por la que se descartó trabajar con velocidades, debido a que las distancias entre los nodos que conforman a los SIG tanto de las líneas del metro como los SIG de las avenidas principales de la CDMX no son equidistantes y en algunas secciones están conglomerados, mientras que en otras secciones hay muy pocos nodos.

Sin duda una de lecciones más importantes al momento de realizar este trabajo fue de cómo yendo de menos a más se facilita la modelación, es decir, al inicio se abordó el problema en su máxima complejidad, tratando de abarcar todo al mismo tiempo, esto no sólo resultó en un proceso de modelaje frustrante y lento, sino que lleno de errores en la implementación, fue solamente hasta que se hizo el cambio de perspectiva de hacer modelos pequeños para construir modelos más grandes que se tuvo avances más rápidos en el modelo. Se hicieron varios modelos pequeños que evolucionaron y sirvieron de base para el modelo final, por ejemplo se hizo un modelo pequeño a partir del modelo de (Gil Álvaro, 2012) sobre el algoritmo Dijkstra para una red de seis nodos variando las condiciones hasta que fueran apropiadas para el modelo de una línea, de la misma forma se hizo un modelo del sub-modelo de una línea para experimentar con la extensión de “networks” de NetLogo™, así para algunas secciones donde resultaba más conveniente probar una sección de código en una red pequeña compuesta por cinco a diez agentes para luego implementar ese código a la red de las líneas del metro y avenidas principales.

Similar el punto anterior, mantener la simplicidad en las reglas del modelo como primer paso fue crucial para el avance en el desarrollo del modelo, en repetidas ocasiones se intentó codificar un comportamiento complejo que resultaba con errores o no se comportaba de la manera deseada. Iniciar con reglas sencillas e ir añadiendo complejidad fue sin duda una forma muy efectiva de desarrollar el modelo, así fue como se desarrolló el movimiento de los trenes, primero

asegurándose que se logaran mover de un enlace direccionado a otro, luego que no se regresaran al enlace anterior, luego implementando el incidente, seguido implementar el cambio de sentido sobre la misma línea y así sucesivamente hasta tener el comportamiento total.

Los cambios en el enfoque de modelación sin duda es algo que impactó directamente tanto en el modelo conceptual como en el modelo computacional, es decir, su implementación. Inicialmente se contemplaba la interacción de ambulancias y camiones de bomberos con coches en avenidas principales pero se descartó debido al tiempo de cómputo y la complejidad del movimiento de las ambulancias y su interacción con el tráfico de la Ciudad de México, se prefirió obtener la mayor cantidad de rutas más cortas en poco tiempo a observar pocas rutas por darle prioridad a la interacción y desplazamiento de las ambulancias y camiones de bomberos.

Las ventajas y desventaja de trabajar con nodos para el movimiento de agentes tipo tren en NetLogo™. Entre las ventajas fue posible el uso del algoritmo de Dijkstra para obtener la ruta más corta ya que es un algoritmo de redes, se revisó la posibilidad de trabajar con “patches” o parches (agente estacionario), ya que existen modelos de tráfico utilizando parches y funcionan muy bien introduciendo velocidades, sin embargo, para las dimensiones geográficas que se querían abarcar de la Ciudad de México el utilizar parches era menos viable, ahora una desventaja de utilizar nodos y enlaces es que no se pueden establecer velocidades debido a que no hay un control en la distancia y número de nodos, estos vienen dados por los datos del GIS, entonces un problema surgió cuando había más de 10 nodos traslapados y formando enlaces entre sí, es por este problema que se simplificó la red y esta simplificación hace que las velocidades aún sean menos viables, además en muchos casos se generaba un movimiento poco realista de autos, ambulancias y camiones de bomberos, tanto por la cantidad de nodos traslapados previo a la simplificación como por la distancia recorrida nodo a nodo visualmente parecía que aceleraba a pesar de que se moviera un solo nodo por tic esto luego de hacer la simplificación. Sin embargo, este no fue el caso de las redes de las líneas de metro, ya que se mantuvieron como estaban en los archivos .shp, dándoles una relación de 1:1 con el sistema real, aunque, esto también tuvo un impacto directo en la “velocidad” de los trenes, que parece ser variable pero esto depende de cómo se muestrearon los nodos del SIG.

Algunas limitaciones que tiene el modelo de simulación-optimización de este trabajo vienen dadas por las condiciones de NetLogo™, más propiamente por su escala, ya que la relación al número de nodos por parches es muy importante, durante los experimentos se configuró el mundo de NetLogo™ para que fuera de (31*31) esto para obtener datos rápidamente al disminuir los recursos computacionales, sin embargo, si se incrementa el tamaño del mundo se obtienen resultados con distancias más certeras porque la simplificación de la red es menor, para el comportamiento agregado que se presentó se requeriría una visualización más grande del mundo para obtener datos significativos.

Este trabajo puede ser útil para apoyar la toma de decisiones de las autoridades del STC de la Ciudad de México debido al indicador sobre la cercanía de los servicios de emergencia en cuanto a rutas más cortas, lo que se traduce en atención a emergencias como una herramienta para el posible desarrollo de estrategias y planes para atender las líneas del metro con grandes distancias en rutas más cortas y al indicador sobre la resiliencia de los sistemas de servicios de emergencia relacionados al STC el cual es una herramienta para desarrollar diferentes planes en caso de que se quiera tener un sistema STC-Servicios de Emergencia con mayor resiliencia. Además, este modelo también se podría adaptar y soportar otros sistemas de transporte de tren ubicados en diferentes ciudades siempre y cuando los sistemas de información geográfica estén disponibles.

Las futuras investigaciones son:

- a. La integración de la congestión vehicular en avenidas primarias y secundarias.
- b. La integración de rutas alternativas considerando restricciones espaciales.
- c. La escala de tiempo de simulación y tiempo real.
- d. La implementación de otros algoritmos como el A* y su comparativa con el algoritmo de Dijkstra
- e. La conversión de las distancias obtenidas en pixeles a Km.
- f. La consideración indicadores oficiales a partir de los indicadores obtenidos para la evaluación de la necesidad de servicios de emergencia.

REFERENCIAS

- 1) Brailsford, S., Dangerfield, B., & Churilov, L. (Eds.). (2014). *Discrete-Event Simulation and System Dynamics for Management Decision Making*. Wiley.
- 2) Brailsford, S. C., Eldabi, T., Kunc, M., Mustafee, N., & Osorio, A. F. (2019). Hybrid simulation modeling in operational research: A state-of-the-art review. *European Journal of Operational Research*, 278(3), 721-737. <https://doi.org/10.1016/j.ejor.2018.10.025>.
- 3) Calabrò, G., Inturri, G., Le Pira, M., Pluchino, A., & Ignaccolo, M. (2020). Bridging the gap between weak-demand areas and public transport using an ant-colony simulation-based optimization. *Transportation Research Procedia*, 45(1), 234-241. 10.1016/j.trpro.2020.03.012
- 4) Chen, P.-H., & Feng, F. (2009). A fast flow control algorithm for real-time emergency evacuation in large indoor areas. *Fire Safety Journal*, 44(5), 732-740. 10.1016/j.firesaf.2009.02.005
- 5) Chen, S., Di, Y., Liu, S., & Wang, B. (2017). Modeling and Analysis on Emergency Evacuation from Metro Stations. *Mathematical Problems in Engineering*, 2017, 11. <https://doi.org/10.1155/2017/2623684>
- 6) Chen, S., Zhang, Y., Di, Y., Li, F., & Jia, W. (2018). A MULTI-AGENT-BASED APPROACH FOR THE IMPACTS ANALYSIS OF PASSENGER FLOW ON PLATFORMS IN METRO STATIONS CONSIDERING TRAIN OPERATIONS. *Transport*, 33(3), 821-834. <https://doi.org/10.3846/transport.2018.5663>
- 7) China Urban Rail Transit Market. (2021, 08 18). International Trade Administration. Retrieved March 10, 2022, from <https://www.trade.gov/market-intelligence/china-urban-rail-transit-market>
- 8) Choi, J., Kim, S.-G., Lee, J., & Choi, Y. S. (2013). Agent-Based Evacuation Simulation for Building Structure Evaluation. *GIScience & Remote Sensing*, 46(4), 347-364. <http://dx.doi.org/10.2747/1548-1603.46.4.347>
- 9) Comisión de Derechos Humanos del Distrito Federal (2017) *DIAGNÓSTICO SOBRE EL SERVICIO Y LAS INSTALACIONES DEL SISTEMA DE TRANSPORTE COLECTIVO 2013-2018* CDHDF México
- 10) CONAPO, INEGI & SEDESOL (2004). Delimitación de las zonas metropolitanas de México. Secretaría de Desarrollo Social.
- 11) Dangerfield, B. (2014). Systems thinking and system dynamics: A primer. *Discrete-Event Simulation and System Dynamics for Management Decision Making*, 26-51. <https://doi.org/10.1002/9781118762745.ch03>
- 12) Flores de la Mota, I. (2010). *Apuntes teoría de redes*. Primera reimpresión. México, UNAM.

- 13) George Mason University (S.F.) GMU roads model inspired by the Reston commuters model by Melanie Swartz.
- 14) Gil, Alvaro (2012), Application of Kruskal's and Dijkstra's Algorithms with NetLogo. École Polytechnique de Montréal, QC. (alvaro.gil@polymtl.ca)
- 15) Grimm, V. &. (2005). Individual-based modeling and ecology. New Jersey: Princeton University Press.
- 16) Gross, G. (2015). Dispersion scenarios for pollution release in an occupied underground station - a numerical study with a micro-scale and a multi-agent model. *Meteorologische Zeitschrift*, 24(5), 511-524. 10.1127/metz/2015/0680
- 17) Gobierno del Distrito Federal, Secretaría de Desarrollo Social y Gobierno del Estado de México, (1998) *PROGRAMA DE ORDENACIÓN DE LA ZONA METROPOLITANA DEL VALLE DE MÉXICO* <https://paot.org.mx/centro/programas/POZM.pdf>
- 18) Hu, H., Gao, Y., & Liu, Z. (2020). Integrated Planning and Operation of Bus Bridging Evacuation for Metro Rail Disruption. *American Society of Civil Engineers*. 146(4), 1-9. [https://doi.org/10.1061/\(ASCE\)UP.1943-5444.0000625](https://doi.org/10.1061/(ASCE)UP.1943-5444.0000625)
- 19) Huang, C. Y., Lai, C. Y., & Cheng, K. T. (2009). Fundamentals of Algorithms. In *Electronic Design Automation* (pp. 173–234). Elsevier Inc. <https://doi.org/10.1016/B978-0-12-374364-0.50011-4>
- 20) Huang, W., Zhou, B., Yu, Y., & Sun, H. &. (2021). Using the disaster spreading theory to analyze the cascading failure of urban rail transit network. *Sichuan: Reliability Engineering and System Safety*.
- 21) INEGI. (17 de Septiembre de 2021). Censos y Conteos de Población y Vivienda. Obtenido de Censos y Conteos de Población y Vivienda: <https://www.inegi.org.mx/programas/ccpv/2020/default.html>
- 22) INEGI. (14 de Octubre de 2021). Encuesta Origen Destino en Hogares de la Zona Metropolitana del Valle de México (EOD) 2017. Obtenido de <https://www.inegi.org.mx/programas/eod/2017/>
- 23) Kent State Online (01 de junio de 2021). *What are the five components of GIS?* Recuperado el 07 septiembre 2022 de <https://onlinedegrees.kent.edu/geography/geographic-information-science/community/what-are-the-five-components-of-gis>
- 24) Kraus, L., Stanojevic, M., Tomašević, N., & Mijovic, V. (2011). A Decision Support System for Building Evacuation based on the EMILI SITE environment. 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 334-336. 10.1109/WETICE.2011.59
- 25) Li, Z., Tang, M., Liang, D., & Zhao, Z. (2016). Numerical simulation of evacuation in a subway station. *Procedia Engineering*, 135, 615-320. 10.1016/j.proeng.2016.01.126

- 26) Lo, S.M., Wang, W.L., Liu, S.B., & Ma, J. (2014). Using Agent-based Simulating Model for Studying Fire Escape Process in Metro Stations. *Procedia Computer Science*, 32, 388-396. 10.1016/j.procs.2014.05.439
- 27) Machi, L. &. (2009). *The Literature Review*. USA: Corwin Press.
- 28) QGIS. (S.F.). Acerca de QGIS. Descubre QGIS. Recuperado el 16 de Marzo, 2022, de <https://www.qgis.org/es/site/about/index.html>
- 29) Rand, W., & Rust, R. T. (2011). Agent-based modeling in marketing: Guidelines for rigor. *International Journal of Research in Marketing*, 28 (3), 181 – 193.
- 30) Salinas-Arreortua, L. (2017). Gestión metropolitana en la Zona Metropolitana del Valle de México: entre la legalidad y la voluntad política. *Papeles de población*, 23(91), 143-169. <https://doi.org/10.22185/24487147.2017.91.007>
- 31) Secretaría de Movilidad. (28 enero 2021). *Vialidades de la Ciudad de México*. Portal de Datos Abiertos del Gobierno de la Ciudad de México. Recuperado el 24 enero 2022 de: <https://datos.cdmx.gob.mx/dataset/vialidades-de-la-ciudad-de-mexico/resource/44bc1807-a7ba-44a0-bf46-504623a30905>
- 32) Secretaría de Salud. (12 enero 2021). *Hospitales ISSTE: Hospitales ISSTE de la Ciudad de México*. Portal de Datos Abiertos del Gobierno de la Ciudad de México. Recuperado el 24 enero 2022 de: <https://datos.cdmx.gob.mx/dataset/hospitales-y-centros-de-salud>
- 33) Secretaría de Salud. (12 enero 2021). *Hospitales y Centros de Salud: Ubicaciones georreferenciadas de los hospitales y Centros de Salud del Gobierno de la Ciudad de México*. Portal de Datos Abiertos del Gobierno de la Ciudad de México. Recuperado el 24 enero 2022 de: <https://datos.cdmx.gob.mx/dataset/hospitales-y-centros-de-salud>
- 34) Sistema de Transporte Colectivo STC. (10 de Septiembre de 2021). Características generales del material rodante. Obtenido de <https://metro.cdmx.gob.mx/operacion/mas-informacion/material-rodante>
- 35) Sistema de Transporte Colectivo STC. (10 de Septiembre de 2021). Conformación de un Tren del Metro. Obtenido de <https://www.metro.cdmx.gob.mx/operacion/conformacion-de-un-tren>
- 36) Sistema de Transporte Colectivo STC. (2017). *DIAGNÓSTICO SOBRE EL SERVICIO Y LAS INSTALACIONES DEL SISTEMA DE TRANSPORTE COLECTIVO 2013-2018*. CDMX: STC.
- 37) Sistema de Transporte Colectivo STC. (10 de Septiembre de 2021). *FUNCIONAMIENTO DEL CARRO*. Obtenido de <https://www.metro.cdmx.gob.mx/operacion/carro-neumatico-y-ferrero>

- 38) Sistema de Transporte Colectivo STC. (1 de Septiembre de 2021). Mapa de Red del Metro. Obtenido de https://www.metro.cdmx.gob.mx/storage/app/media/red/plano_red19.pdf
- 39) Sistema de Transporte Colectivo STC. (20 de Septiembre de 2022). Parque Vehicular. Obtenido de <https://www.metro.cdmx.gob.mx/parque-vehicular>
- 40) Sistema de Transporte Colectivo STC. (2018). PLAN MAESTRO DEL METRO 2018-2030. CDMX: STC.
- 41) Sistema de Transporte Colectivo STC. (12 enero 2021). *Ubicación de líneas y estaciones del Sistema de Transporte Colectivo Metro: Geolocalización de líneas y estaciones del Sistema de Transporte Colectivo Metro*. Portal de Datos Abiertos del Gobierno de la Ciudad de México. Recuperado el 24 enero 2022 de: <https://datos.cdmx.gob.mx/dataset/lineas-y-estaciones-del-metro>
- 42) Stubenschrott, M., Matyus, T., Schrom-Feiertag, H., Kogler, C., & Seer, S. (2017). Route-Choice Modeling for Pedestrian Evacuation Based on Infrastructure Knowledge and Personal Preferences. *Transportation Research Record: Journal of the Transportation Research Board*, 2623(1), 82-89. <http://dx.doi.org/10.3141/2623-09>
- 43) USGS (S.F.). *What is geographic information system (GIS)?* Recuperado el 07 septiembre 2022 de <https://www.usgs.gov/faqs/what-geographic-information-system-gis>
- 44) GISGeography (20 de junio de 2022). *What is a Geographic Information System (GIS)?* GISGeography Recuperado el 07 septiembre 2022 de <https://gisgeography.com/what-gis-geographic-information-systems/>
- 45) Wang, F. (2021). Multi-scenario simulation of subway emergency evacuation based on multi-agent. *Int j simul model*, 20(2), 387-397. <https://doi.org/10.2507/IJSIMM20-2-CO8>
- 46) Wang, J., Guo, J., Wu, X.-M., & Guo, X.-H. (2020, Septiembre). Study on intelligent algorithm of guide partition for emergency evacuation of a subway station. *IET Intelligent Transport Systems*, 14(11), 1440-1446. 10.1049/iet-its.2019.0655
- 47) Wilensky, U. & William, R. (2015). *An Introduction to Agent-Based Modeling Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. Massachusetts: MIT Press.
- 48) Wilensky, U. 1999. NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
- 49) Wu, C.-H., & Chen, L.-C. (2013). The Application of Algorithms for Fire-fighting Search Operations within Buildings. *International Conference on Digital Manufacturing and Automation (ICDMA)*, 4(1), 1383-1387. 10.1109/ICDMA.2013.330

- 50) Xi, Y., Du, Q., He, B., Ren, F., Zhang, Y., & Ye, X. (2017). The Dynamic Optimization of the Departure Times of Metro Users during Rush Hour in an Agent-Based Simulation: A Case Study in Shenzhen, China. *Applied Sciences*, 7(11), 1102. <http://dx.doi.org/10.3390/app7111102>
- 51) Xie, J., Chen, K., Kwan, T. H., & Yao, Q. (2020). Numerical simulation of the fire emergency evacuation for a metro platform accident. *Simulation: Transactions of the Society for Modeling and Simulation International*, 97(1), 1-14. <https://doi.org/10.1177/0037549720961433>
- 52) Yin, H., Wu, J., Liu, Z., Yang, X., & Qu, Y. (2019). Optimizing the release of passenger flow guidance information in urban rail transit network via agent-based simulation. *Applied Mathematical Modelling*, 72, 337-355. <https://doi.org/10.1016/j.apm.2019.02.003>.

ANEXO A: MODELACIÓN EN NETLOGO™

A continuación se describe una estructura general y algunos de los elementos que conforman el modelo de simulación basado en agentes, desarrollado en NetLogo™. NetLogo™ es secuencial por lo tanto hay procedimientos que se tienen que definir previamente. Conviene revisar el diccionario de NetLogo™ para entender la sintaxis y función de primitivos que se mencionan y también de aquellos que no se mencionan. El diccionario puede ser abierto mediante la pestaña de ayuda en el programa o también se encuentra en los archivos de instalación de este.

- Primero se habilitan extensiones si se desea usarlas

```
extensions [
```

```
csv
```

```
gis
```

```
array
```

```
table
```

```
sample
```

```
sound
```

```
]
```

- Siguiendo se definen las familias (razas) de agentes o links (direccionados, no direccionados)

```
breed [familia integrante_de_familia]
```

```
directed-link-breed [familia integrante_de_familia]
```

```
undirected-link-breed[familia integrante_de_familia]
```

- Se define el listado de variables globales, estas variables permanecen a lo largo de todo el código, también se pueden definir variables locales, sin embargo, esas sólo permanecen en la función en la que definen

```
globals [
```

```
Variable global 1
```

Variable global 2

...

Variable global n

]

- Se definen las propiedades de los agentes (peso, velocidad, pasajeros etc.)

turtles-own [

Propiedad 1

Propiedad 2

...

Propiedad n

]

- Para establecer las condiciones iniciales del modelo, se define un procedimiento “setup” el cual puede activarse por el usuario mediante un botón o también puede configurarse para que se active al abrir el modelo

to setup

clear-all (este primitivo permite limpiar todos los procesos previos)

Procedimiento 1 de condiciones iniciales

Procedimiento 2 de condiciones iniciales

...

Procedimiento n de condiciones iniciales

end

- En ocasiones es útil definir una función por separado las condiciones iniciales de los agentes, esta función puede ser posteriormente incluida en el “setup”

to setup-turtles

set propiedad valor inicial

set color cyan; asigna un color set size 3; asigna el tamaño set shape "shape" ;
asigna la forma setxy 0 0 ; coordenadas x y

set heading grados ;

end

- Para la ejecución del modelo principal, se define el botón “to go”

to go

Procedimiento 1 de condiciones iniciales Procedimiento 2 de condiciones
iniciales ...

Procedimiento n de condiciones iniciales

End

En ocasiones se requiere crear, abrir, y/o modificar un archivo csv, ya sea para guardar datos que produce el modelo o para que el modelo lea datos para su funcionamiento. Se requiere habilitar la extensión csv.

- Preparación de archivos csv

to setup_file_csv

file-close-all ; (Cierra todos los archivos)

reset-ticks; (pone en cero al contador del tiempo)

end

- Escritura de archivo csv

to write- to-csv

csv:to-file "archivo.csv" [(list propiedad 1 propiedad 2 propiedad n)] of turtles

end

- Lectura de archivo csv

to read- from-csv

file-close-all ; Cierra todos los archivos

f not file-exists? "archivo.csv" [

ser-message "No file 'archivo.csv' exists! “

```

top
]
file-open "archivo.csv" ; abre el archivo de lectura
while [ not file-at-end? ] [
let data csv:from-row file-read-line
create-turtles 1 [
set xcor item 0 data
set ycor item 1 data
set size item 2 data
set color item 3 data
set shape "forma"
]
file-close ; se cierran todos los archivos
end

```

También se pueden cargar archivos .shp de sistemas de información geográfica (GIS), se puede cargar los mapas cuando se necesiten o definir una función que cargue los mapas y llamar la función cuando se necesite. Se recomienda hacer lo posterior.

- Preparación de mapas para cargar

```

to setup_maps

gis:load-coordinate-system (word "Carpeta/"projection".prj")

set dataset gis:load-dataset "Carpeta/archivo.shp" gis:set-world-envelope
gis:envelope-of Mexico-dataset

end

```

- Dibujar mapas (como imagen sobre la interfaz del modelo)

```

to display-Maps

gis:set-drawing-color código

```

```
gis:fill dataset 1
gis:set-drawing-color código
gis:draw dataset 3
end
```

- Exportar datos de salida

```
to save-outputs
```

```
export-all-plots plot-file
```

```
export-world "Archivo.csv"
```

```
file-close-all
```

```
end
```

- Imprimir en imagen el área de trabajo

```
to-save-world
```

```
export-view (word "name-of-the-file" ticks ".png" ) end
```

Diseño de interfaz de usuario EJECUTAR

- Botón. Mediante un botón podemos ejecutar los PROCEDIMIENTOS que se desarrollan en el código. El botón requiere únicamente el nombre del PROCEDIMIENTO. Puede ejecutarse una sola vez o bien continuamente.

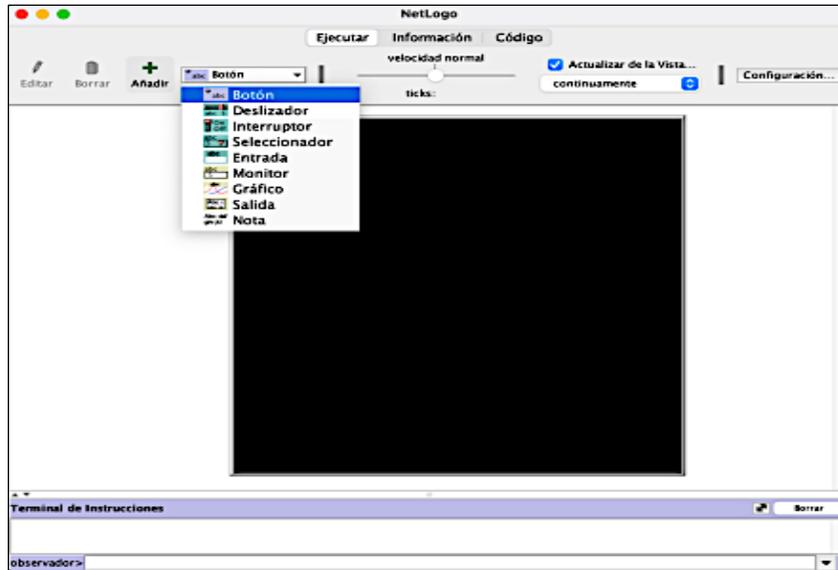


Figura A.1 Interfaz del software NetLogo™, modo de insertar Botones.
 Fuente: Elaboración propia (2022).

- Deslizador. El deslizador se utiliza para que el usuario del modelo fije los valores de variables globales. Al ser un deslizador incluye el valor mínimo, máximo y por default de la variable global en cuestión.

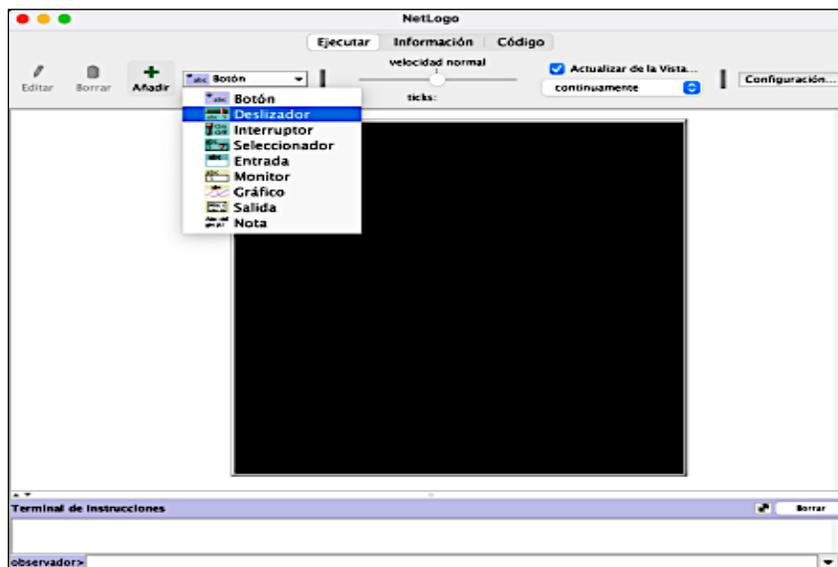


Figura A.2 Interfaz del software NetLogo™, modo de insertar Deslizadores.
 Fuente: Elaboración propia (2022).

- Interruptor (ON/OFF). El interruptor permite habilitar/deshabilitar variables. En el código se verifica el estado de la variable y según corresponda a habilitada/deshabilitada se ejecutan ciertas acciones.

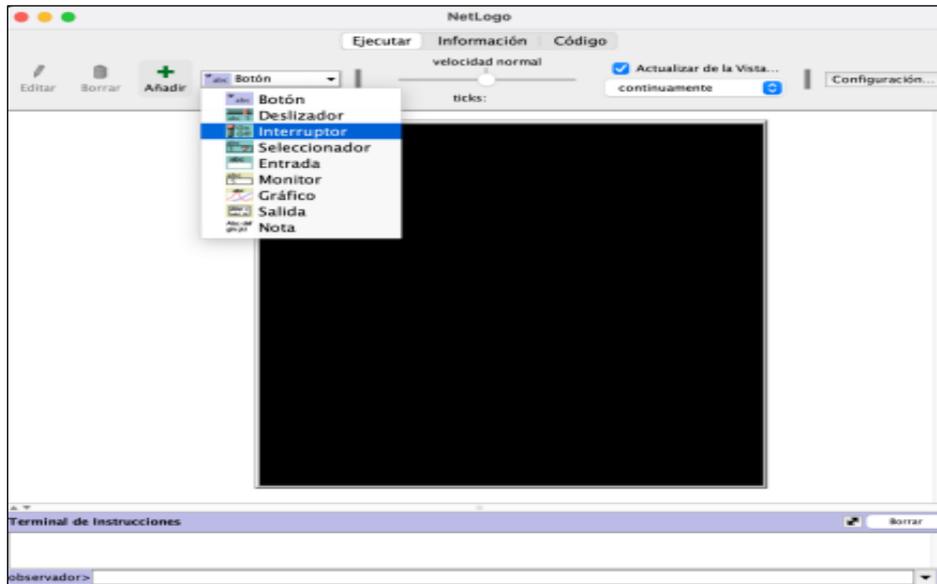


Figura A.3 Interfaz del software NetLogo™, modo de insertar Interruptores.
Fuente: Elaboración propia (2022).

- Seleccionador. Permite que el usuario del modelo seleccione desde un menú el valor de una variable.

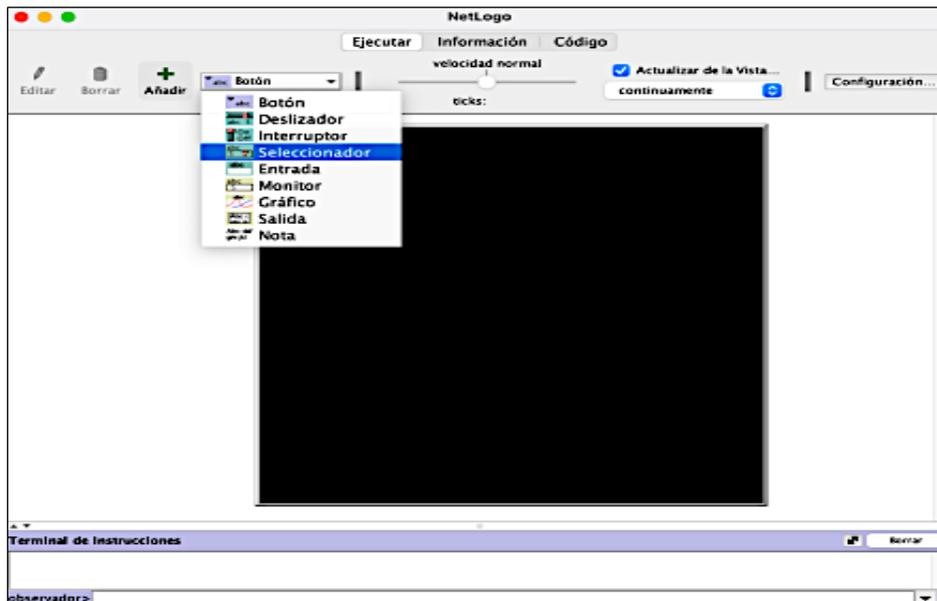


Figura A.4 Interfaz del software NetLogo™, modo de insertar Seleccionadores.
Fuente: Elaboración propia (2022).

- Entrada. Habilita las entradas del modelo en diferentes formatos. El usuario del modelo debe escribir los valores pertinentes.

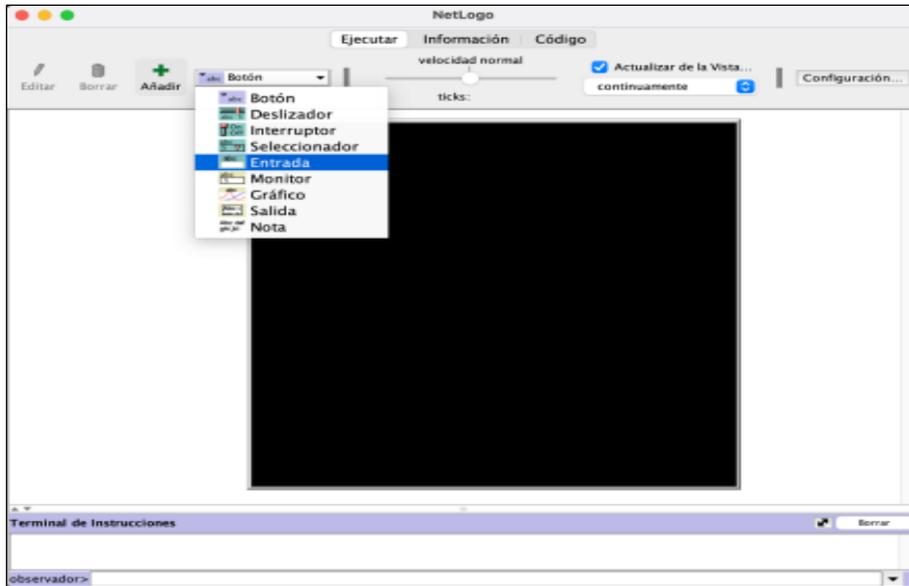


Figura A.5 Interfaz del software NetLogo™, modo de insertar Entradas de Datos.
Fuente: Elaboración propia (2022).

- Monitor. Posibilita el monitoreo de variables cuando el código se está ejecutando.

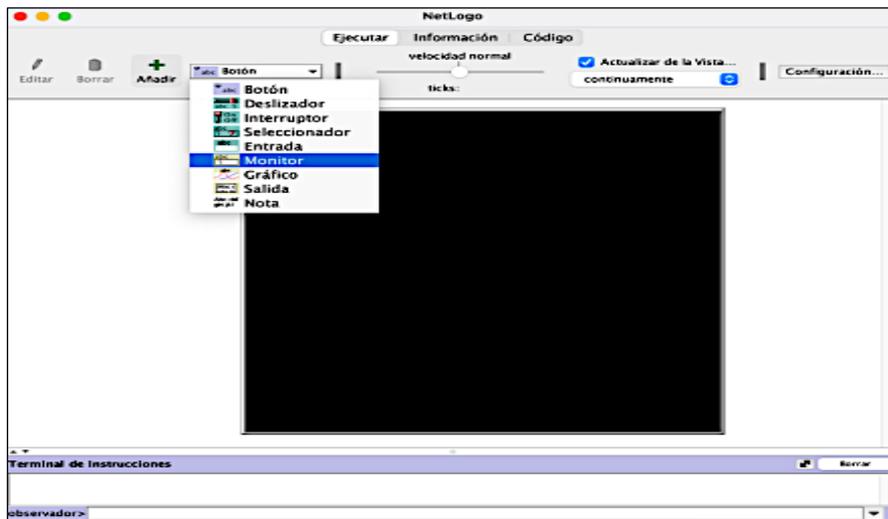


Figura A.6 Interfaz del software NetLogo™, modo de insertar Monitores.
Fuente: Elaboración propia (2022).

- Gráfico. Posibilita el monitoreo de variables de salida en formato de gráfico. Pueden diseñarse histogramas.

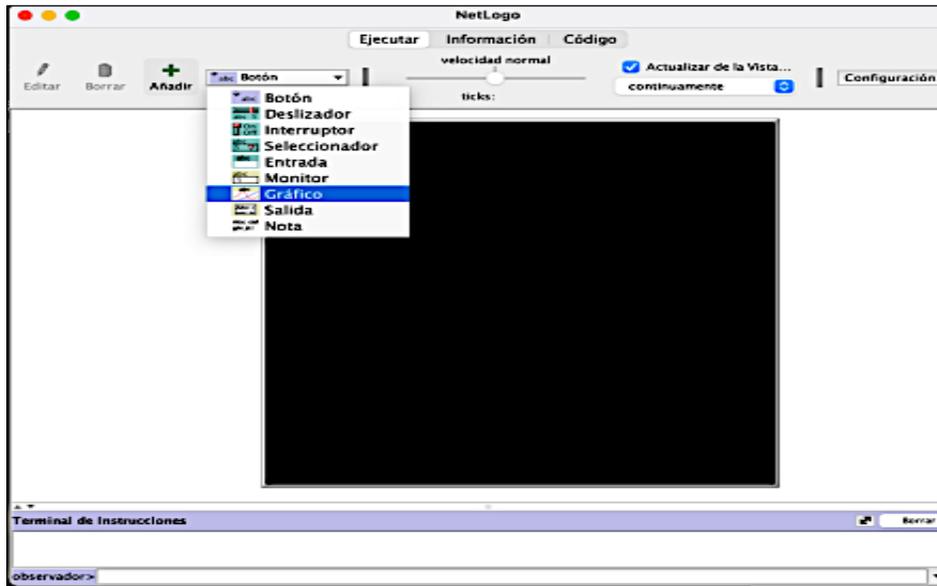


Figura A.7 Interfaz del software NetLogo™, modo de insertar Gráficos.
Fuente: Elaboración propia (2022).

- Salida. Muestra una salida del modelo en formato de texto. Sirve para apoyar el envío de mensajes al usuario final.

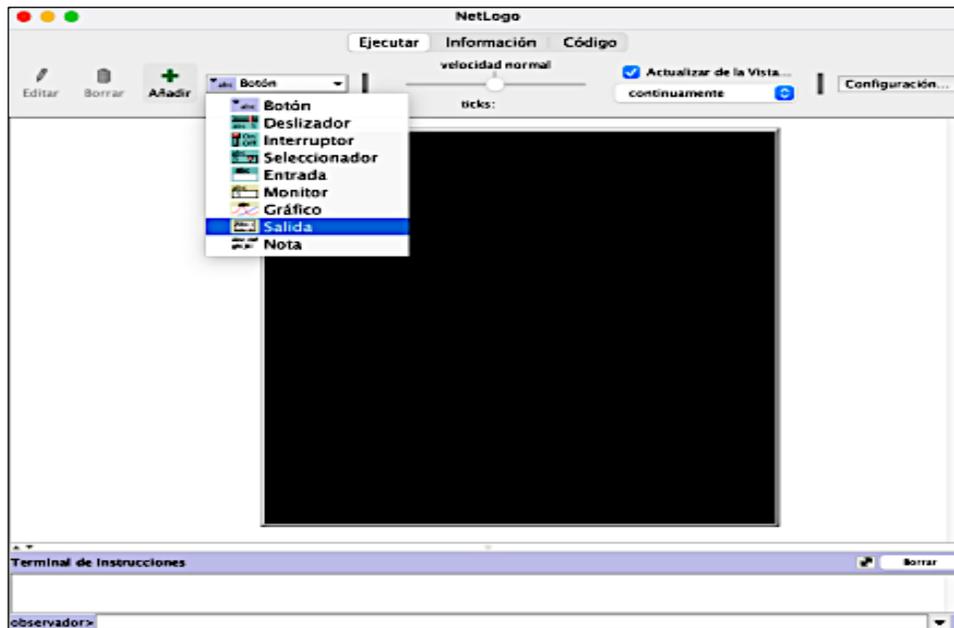


Figura A.8 Interfaz del software NetLogo™, modo de insertar Salidas de Datos.
Fuente: Elaboración propia (2022).

- Nota. A través de una Nota puede introducirse texto en la interfaz del usuario final.

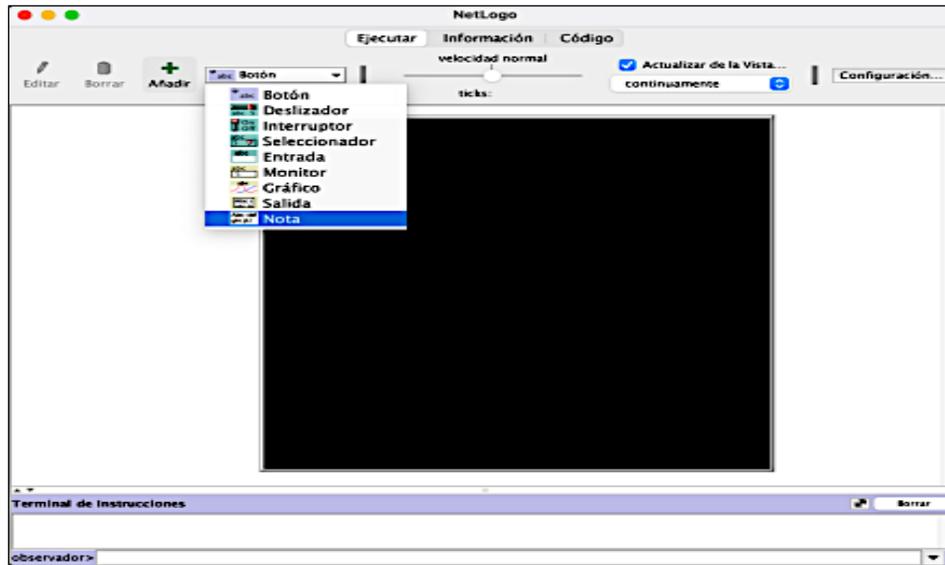


Figura A.9 Interfaz del software NetLogo™, modo de insertar Notas de Texto.
Fuente: Elaboración propia (2022).

ANEXO B: EXPERIMENTO DE PROYECCIONES

Se realizó un experimento extra considerando un cambio de proyección cartográfica (Cónica de Lambert, WGS 85 y US Orthographic) para la línea 1, aquí se reportan los resultados. Este experimento adicional se realizó para conocer y comparar si la selección de una proyección en particular tenía relevancia significativa sobre las distancias en pixeles obtenidas. Se invita al lector a extraer sus propias conclusiones, observando los valores de distancia media hospital y distancia media bomberos.

Tabla B.1 Experimento de proyecciones L1 escenario: IID.
Fuente: Elaboración propia (2022).

BehaviorSpace results (NetLogo 6.2.2)						
Modelo_Tesis.nlogo						
Experimento_proyecciones_L1_IID						
08/31/2022 16:13:34:315-0500						
	min-pxcor	max-pxcor	min-pycor	max-pycor		
[run number]	-15	15	-15	15		
Projection	1	2	3	4	5	6
Projection	WGS_84_Geographic	WGS_84_Geographic	US_Orthographic	US_Orthographic	Lambert_Conformal_Conic	Lambert_Conformal_Conic
show_Mapas?	true	true	true	true	true	true
show_Avenidas_Principales?	true	true	true	true	true	true
number-of-trenes	8	8	8	8	8	8
usar_rutina_Dijkstra?	false	false	false	false	false	false
probabilidad_corto_circuito	1	1	1	1	1	1
probabilidad_heridos	50	50	50	50	50	50
probabilidad_incendio	50	50	50	50	50	50
show_Bomberos?	true	true	true	true	true	true
show_ISSSTE?	true	true	true	true	true	true
show_Centros_Salud?	true	true	true	true	true	true
ISSSTE_Resources	low	low	low	low	low	low
Centros_Salud_Resources	low	low	low	low	low	low
Fire_Station_Resources	low	low	low	low	low	low
[reporter]	IncidentesL1	IncidentesL1	IncidentesL1	IncidentesL1	IncidentesL1	IncidentesL1
[final]	1000	1000	1000	1000	1000	1000
[min]	0	0	0	0	0	0
[max]	1000	1000	1000	1000	1000	1000
[mean]	502.5488508	507.7676577	487.8307225	490.4045362	507.6627531	491.4060327
[steps]	12660	12175	12830	12344	12690	12796
[reporter]	IncidentesAmbulancias	IncidentesAmbulancias	IncidentesAmbulancias	IncidentesAmbulancias	IncidentesAmbulancias	IncidentesAmbulancias
[final]	488	534	498	537	488	500
[min]	0	0	0	0	0	0
[max]	488	534	498	537	488	500
[mean]	243.1269252	274.5686597	237.7557478	268.9300932	254.4905051	250.2206767
[steps]	12660	12175	12830	12344	12690	12796
[reporter]	IncidentesBomberos	IncidentesBomberos	IncidentesBomberos	IncidentesBomberos	IncidentesBomberos	IncidentesBomberos
[final]	475	497	493	502	512	477
[min]	0	0	0	0	0	0
[max]	475	497	493	502	512	477
[mean]	231.3842508	248.5050099	237.7887928	245.6266505	256.9363328	236.5640384
[steps]	12660	12175	12830	12344	12690	12796
[reporter]	distancia_media_hospital_L1	distancia_media_hospital_L1	distancia_media_hospital_L1	distancia_media_hospital_L1	distancia_media_hospital_L1	distancia_media_hospital_L1
[final]	7.112577767	7.309064013	7.581704017	6.834295862	7.668630498	7.407979127
[min]	0	0	0	0	0	0
[max]	11.4907636	8.434103492	9.361692709	7.145245875	13.48890268	11.6794541
[mean]	7.135687424	7.301889791	7.515890451	6.618090969	7.470751647	7.550120529
[steps]	12660	12175	12830	12344	12690	12796
[reporter]	distancia_media_bom_L1	distancia_media_bom_L1	distancia_media_bom_L1	distancia_media_bom_L1	distancia_media_bom_L1	distancia_media_bom_L1
[final]	9.035239411	8.83187022	8.836093683	8.950644907	8.966090325	8.404915764
[min]	0	0	0	0	0	0
[max]	10.60778427	16.17475442	9.517458525	9.179910589	9.36868013	8.925747659
[mean]	9.132030747	9.103131619	8.555037961	8.594577392	8.80469426	8.372205216
[steps]	12660	12175	12830	12344	12690	12796