



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

DESARROLLAR UN MODELO BASADO EN AGENTES
DE VALENCIA Y SU AREA METROPOLITANA
UTILIZANDO MATSIM

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

AUTOR/A: Noguera Urios, Martin Sigfrido

Tutor/a: Ballester Merelo, Francisco José

Cotutor/a: Arroyo López, María Rosa

CURSO ACADÉMICO: 2023/2024



RESUMEN:

Este proyecto de modelización de la movilidad en Valencia y su área metropolitana usando MATSim, se enfoca en la configuración técnica y metodológica necesaria para llevar a cabo simulaciones precisas. Se describe la creación de un entorno de desarrollo que integra herramientas como Java, Maven, y GitHub, así como el desarrollo de algoritmos para la conversión de datos a formatos adecuados para MATSim. Se pone énfasis en la recopilación, tratamiento y conversión de datos, destacando el uso de software complementario para la visualización de resultados. Además, se menciona la implementación de políticas de movilidad mediante la simulación de escenarios, subrayando cómo la metodología basada en agentes es utilizada en el contexto del estudio. El proyecto presenta técnicas y estrategias para la simulación detallada de la movilidad, ofreciendo una base sólida para futuras investigaciones y aplicaciones en el campo de la planificación del transporte.

ABSTRACT:

This mobility modelling project in Valencia and its metropolitan area using MATSim focuses on the technical and methodological setup necessary to carry out accurate simulations. It describes the creation of a development environment that integrates tools such as Java, Maven, and GitHub, as well as the development of algorithms for the conversion of data into formats suitable for MATSim. Emphasis is placed on data collection, processing and conversion, highlighting the use of complementary software for the visualisation of results. Furthermore, the implementation of mobility policies through scenario simulation is mentioned, highlighting how agent-based methodology is used in the context of the study. The project presents techniques and strategies for detailed mobility simulation, providing a solid basis for future research and applications in the field of transport planning.

RESUM:

Este projecte de modelització de la mobilitat a València i la seua àrea metropolitana usant MATSim, s'enfoca en la configuració tècnica i metodològica necessària per a dur a terme simulacions precises. Es descriu la creació d'un entorn de desenvolupament que integra ferramentes com Java, Maven, i GitHub, així com el desenvolupament d'algorismes per a la conversió de dades a formats adequats per a MATSim. Es posa èmfasi en la recopilació, tractament i conversió de dades, destacant l'ús de programari complementari per a la visualització de resultats. A més, s'esmenta la implementació de polítiques de mobilitat mitjançant la simulació d'escenaris, subratllant com la metodologia basada en agents és utilitzada en el context de l'estudi. El projecte presenta tècniques i estratègies per a la simulació detallada de la mobilitat, oferint una base sòlida per a futures investigacions i aplicacions en el camp de la planificació del transport.

RESUMEN EJECUTIVO:

CONCEPT (ABET)	CONCEPTO (traducción)	¿Cumple?(S/N)	¿Dónde?(páginas)
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	7
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	11-12
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	9-10-11
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	15 a 38
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	11-12
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	67-68
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	68



AGRADECIMIENTOS:

Quisiera expresar mi más profundo agradecimiento a todas aquellas personas que han contribuido a la realización de este estudio y que han estado a mi lado durante todo el proceso.

En primer lugar, mi gratitud a mi familia, quienes han sido mi pilar fundamental a lo largo de mi carrera. Su apoyo incondicional en los momentos más difíciles y su confianza en mis capacidades han sido esenciales para superar los desafíos que he enfrentado. Su amor y paciencia son la base sobre la que he podido edificar mis sueños y aspiraciones.

A mis amigos, les agradezco por ser parte crucial de mi desarrollo académico y personal. Su presencia ha enriquecido mi experiencia universitaria; gracias a ellos he podido mantener un hábito de estudio constante y efectivo. Su amistad ha sido un regalo invaluable que ha hecho este viaje mucho más gratificante.

Un agradecimiento especial a mi tutor Francisco José Ballester Merelo y mi cotutora María Rosa Arroyo López (directora de mi proyecto de prácticas curriculares y extracurriculares), cuya guía y apoyo han sido determinantes en mi formación. Francisco, en particular, merece una mención especial por su apoyo constante desde el primer año de carrera. Su capacidad para tener esa cercanía tan propia suya hacia sus estudiantes y eliminar las barreras tradicionales alumno-profesor han sido fundamentales. Su pasión y dedicación no solo han fortalecido mi amor por este grado, sino que también han inspirado en mí un compromiso duradero con mi campo de estudio.

Finalmente, a todos los profesores que han sido parte de esta etapa de mi vida, gracias por compartir su conocimiento y experiencia. Cada clase, cada proyecto y cada conversación ha sido un paso más hacia este logro.

A todos, gracias de corazón por acompañarme en este camino lleno de aprendizaje y crecimiento.

ÍNDICE

INTRODUCCIÓN	7
1.1 CONTEXTO Y JUSTIFICACIÓN	7
1.2 RELACIÓN DEL PROYECTO CON LOS ODS	8
1.3 OBJETIVO GLOBAL	9
2. OBJETIVOS ESPECÍFICOS	10
3. MARCO TEÓRICO	11
3.1 MODELIZACIÓN DE LA MOVILIDAD	11
• <i>¿Qué es la modelización del transporte?</i>	<i>11</i>
• <i>¿Qué métodos se emplean para la modelización del transporte?</i>	<i>12</i>
3.2 MATSIM	14
• <i>Descripción:</i>	<i>14</i>
• <i>Características:</i>	<i>14</i>
• <i>Aplicación en el Contexto de Este Proyecto:</i>	<i>15</i>
4. METODOLOGÍA	15
4.1 CREACIÓN DEL ENTORNO DE DESARROLLO	15
• <i>Instalación de Herramientas Necesarias</i>	<i>15</i>
• <i>Configuración de GitHub</i>	<i>17</i>
• <i>Preparación del Entorno Local:</i>	<i>18</i>
4.2 RECOPIACIÓN, TRATAMIENTO Y CONVERSIÓN DE DATOS PARA EL MODELADO EN MATSIM	24
• <i>Recopilación de Datos</i>	<i>24</i>
• <i>Tratamiento de Datos</i>	<i>25</i>
• <i>Conversión de Datos</i>	<i>25</i>
• <i>4.2.1 Crear la Network sobre la que se va a trabajar</i>	<i>26</i>
• <i>4.2.2 Obtención de config.xml</i>	<i>35</i>
• <i>4.2.3 Transporte público</i>	<i>37</i>
4.3 DESARROLLO DE ALGORITMOS Y HERRAMIENTAS DE CONVERSIÓN	38
• <i>4.3.1 Creación de Scripts XML para la Conversión de Datos</i>	<i>39</i>
• <i>4.3.2 Modificación de Scripts Existente</i>	<i>46</i>
4.4 SOFTWARE COMPLEMENTARIO SIMUNTO VIA	58
• <i>Creación de Simunto Via:</i>	<i>58</i>
• <i>Descripción General:</i>	<i>59</i>
• <i>Características:</i>	<i>59</i>
• <i>Integración con MATSim</i>	<i>60</i>
• <i>Ventajas de Usar SIMUNTO VIA en este Proyecto</i>	<i>61</i>
5. SIMULACIÓN DEL ESCENARIO Y RESULTADOS	62
• INTRODUCCIÓN:	62
• GRABACIONES:	62
6. DISCUSIÓN	66
• APLICABILIDAD DE LA METODOLOGÍA EN OTROS CONTEXTOS:	67
• POSIBLES MEJORAS PARA FUTURAS INVESTIGACIONES:	67
7. CONCLUSIONES	67
• CONCLUSIÓN DEL ESTUDIO:	68
• RECOMENDACIONES:	68
8. BIBLIOGRAFÍA	69

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Objetivos de Desarrollo Sostenible	8
Ilustración 2. Activación osmosis.....	16
Ilustración 3. Repositorio base de GitHub	17
Ilustración 4. Clonación local del repositorio	18
Ilustración 5. Instalación ejecutable simulación (1/2)	20
Ilustración 6. Instalación ejecutable simulación (2/2)	20
Ilustración 7. Ejecutable instalado.....	21
Ilustración 8. Display ejecutable	22
Ilustración 9. Directorio archivos generados	22
Ilustración 10. Extensiones VSCode (1/2)	23
Ilustración 11. Extensiones VSCode (2/2)	24
Ilustración 12. Página web geofabrik.....	26
Ilustración 13. Coordenadas capa centro	27
Ilustración 14. Coordenadas capa norte.....	27
Ilustración 15. Coordenadas capa sur.....	28
Ilustración 16. Coordenadas capa lliria	28
Ilustración 17. Ubicación mapa en formato .pbf	29
Ilustración 18. Recorte capa centro	29
Ilustración 19. Recorte capa norte	29
Ilustración 20. Recorte capa sur.....	30
Ilustración 21. Recorte capa lliria	30
Ilustración 22.m Archivos recortados y en formato .osm.....	30
Ilustración 23. Archivos abiertos en JOSM	31
Ilustración 24. Aspecto archivos en JOSM	32
Ilustración 25. Unión de dos capas	32
Ilustración 26. Plugin MATSim	33
Ilustración 27. Conversión .osm a .xml	34
Ilustración 28. Aspecto network	34
Ilustración 29. Contenido del repositorio inicial	36
Ilustración 30. Config predeterminado.....	36
Ilustración 31. Clonación repositorio de GitHub GTFS2MATSim(1/3)	37
Ilustración 32. Clonación repositorio de GitHub GTFS2MATSim(2/3)	38
Ilustración 33. Clonación repositorio de GitHub GTFS2MATSim(3/3)	38
Ilustración 34. Extensión Maven en VSCode	39
Ilustración 35. Archivos .java de GTFS2MATSim	40
Ilustración 36. Compilación con Maven.....	40
Ilustración 37. Archivos .class generados	41
Ilustración 38. Script RunGTFS2MATSimExample.java (1/2).....	42
Ilustración 39. Script RunGTFS2MATSimExample.java (2/2).....	42
Ilustración 40. Ejecución RunGTFS2MATSimExample.java.....	43
Ilustración 41. Archivos generados	44

INTRODUCCIÓN

1.1 CONTEXTO Y JUSTIFICACIÓN

La modelización de la movilidad urbana, y en particular la modelización basada en agentes a través de MATSim (un software de simulación de transporte basado en agentes), es sumamente importante, ya que se caracteriza por ser una herramienta esencial para profundizar en la comprensión de los sistemas de transporte urbanos densamente poblados, como Valencia y su zona metropolitana.

Esta investigación se sustenta en una metodología avanzada que recrea los patrones de movilidad de los individuos (agentes) con un nivel elevado de detalle, brindando datos valiosos para la planificación y gestión eficiente del transporte. Lo cual no sería posible mediante otras técnicas.

La recolección, tratamiento y conversión adecuada de los datos es fundamental, es un proceso meticuloso que garantiza que los datos no solo sean precisos, sino también relevantes y aplicables a situaciones reales. Además, fortalecer la base de la investigación en ciencias urbanas.

De esta forma, esta introducción presenta las claves para analizar cómo la modelización detallada y el buen tratamiento de los datos pueden tener un impacto positivo en la planificación del transporte en Valencia y establece un precedente metodológico para proyectos futuros en el ámbito de la movilidad urbana.

En las secciones siguientes, se analizarán los métodos empleados para la recolección y procesamiento de datos, la configuración del programa de modelización y los resultados esperados del estudio.

1.2 RELACIÓN DEL PROYECTO CON LOS ODS

Los ODS son las siglas de Objetivos de Desarrollo Sostenible. Son un conjunto de 17 objetivos globales adoptados por los Estados miembros de las Naciones Unidas en septiembre de 2015 como parte de la agenda de 2030 para el Desarrollo Sostenible.



Ilustración 1. Objetivos de Desarrollo Sostenible

Los 17 ODS mostrados en la Figura 1 abordan una amplia gama de cuestiones interconectadas, que incluyen la erradicación de la pobreza, la igualdad de género, la educación de calidad, la acción climática, la paz y la justicia, la reducción de las desigualdades y la promoción de la prosperidad económica, entre otros. Cada objetivo se desglosa en unas metas específicas y se establecen indicadores para medir el progreso hacia su cumplimiento.

Este proyecto puede ser relacionado con varios de estos ODS como:

3. Salud y Bienestar

El estudio contribuye indirectamente a la salud y bienestar al proponer mejoras en la movilidad urbana que pueden reducir la contaminación y el estrés asociados con el tráfico, promoviendo un ambiente urbano más saludable.

9. Industria, Innovación e Infraestructura

La implementación de un modelo avanzado de simulación de transporte basado en agentes como MATSim representa una innovación significativa en la planificación urbana. Esto refuerza el desarrollo de infraestructura eficiente y sostenible, alineada con los objetivos de innovación y mejora de la industria del transporte urbano.

11. Ciudades y Comunidades Sostenibles

El proyecto está directamente relacionado con la creación de ciudades sostenibles mediante la optimización de la movilidad urbana. Las simulaciones buscan mejorar la calidad de vida en áreas urbanas.

13. Acción por el Clima

Al abordar la planificación del transporte y la movilidad urbana, el proyecto contribuirá indirectamente a la reducción de emisiones de gases de efecto invernadero, promoviendo así una acción positiva contra el cambio climático.

17. Alianzas para Lograr los Objetivos

La colaboración entre investigadores y autoridades locales, tal como se menciona en el desarrollo y la implementación de este estudio, refleja la importancia de las alianzas para alcanzar objetivos comunes.

Estos ODS son reflejados en el proyecto a través de la utilización de tecnologías avanzadas y metodologías detalladas para mejorar la planificación y gestión del transporte urbano.

1.3 OBJETIVO GLOBAL

El propósito fundamental de este proyecto es elaborar y aplicar un modelo de simulación de movilidad basado en agentes utilizando MATSim, con el propósito de analizar el sistema de transporte de Valencia.

Se ha llevado a cabo la creación de parte del escenario de MATSim. Partes como puede ser la población o infraestructuras no se han llevado a cabo ya que, creando un número reducido de agentes y de infraestructuras, era suficiente para poder simular los planes necesarios (es decir, para el pleno análisis del estudio).

Así, se va a poder analizar el funcionamiento adecuado, es decir, se establecerán las bases para que otros estudios o proyectos puedan usarlas para evaluar diversas estrategias y/o políticas de transporte. Y que todo esto contribuya a ayudar a implementar el Plan de Movilidad Metropolitano de Valencia [1].

El objetivo fundamental de este estudio es establecer y perfeccionar una metodología para la transferencia de datos (a través de la creación de un entorno local), asegurando que los inputs del modelo sean precisos y significativos para generar resultados fiables y que posibilitan una simulación efectiva de escenarios.

Esto último es fundamental para asegurar la validez y aplicabilidad de los resultados del estudio en la planificación y gestión del transporte urbano.

2. OBJETIVOS ESPECÍFICOS

Este estudio comprende diversos objetivos específicos que guían las etapas operativas del proyecto y garantizan una cobertura exhaustiva de los aspectos necesarios para una simulación efectiva y una evaluación de políticas de movilidad:

- **Delimitación del Área de Estudio (alcance):** El análisis se centrará en la ciudad de Valencia y el área metropolitana, que será la red de transporte, la cual se creará y se determinarán qué tipos de transporte se incluirán. (apartado 4.2.1 Crear la network sobre la que se va a trabajar).
- **Metodología para la Conversión de Datos:** Se Implementará y aplicará una metodología rigurosa para la transferencia de datos existentes a formatos

compatibles con MATSim. Esto posibilitará la integración de diversas fuentes de datos y su aplicación eficaz en la modelización basada en agentes. (apartado 4. Metodología).

- **Recopilación y Tratamiento de Información de Transporte Público:** Se recogerá información actualizada acerca del transporte público utilizando el estándar GTFS, los cuales serán esenciales para representar de manera fiable las paradas y líneas de los servicios de transporte en las simulaciones. (apartado 4.2.3 Transporte Público).
- **Configuración de Herramientas de Visualización:** Se utilizará SIMUNTO VIA como software complementario para examinar los escenarios de simulación y los resultados obtenidos.
- **Implementación Práctica en un Caso de Estudio:** Se aplicará la metodología adoptada en un caso de estudio específico en Valencia, posibilitando, en un futuro, poder validar técnicas y enfoques en un contexto vivencial (ciudad de Valencia).

Este enfoque permite mantener una clara continuidad desde el establecimiento del contexto y la definición del objetivo global, hacia acciones detalladas que describen cómo se logrará dicho objetivo.

3. MARCO TEÓRICO

3.1 MODELIZACIÓN DE LA MOVILIDAD

- **¿Qué es la modelización del transporte?**

Se trata de una investigación digital minuciosa del complejo sistema de transporte y el uso del suelo en el mundo actual. Representa las diversas y complejas opciones de viaje

que ofrecen las personas, sus pautas de desplazamiento y, de esta forma, el nivel de demanda de movilidad, así como la capacidad de la red de sistemas de transporte.

El análisis del transporte es multimodal, ya que se pueden analizar todas las formas de transporte. Esto comprende las bicicletas, los peatones, el transporte público, entre otros. Los modelos de transporte son una representación de la realidad en la que se pueden evaluar digitalmente las consecuencias de las diversas opciones de transporte y uso del suelo.

En consecuencia, la modelización del transporte es, en efecto, una excelente herramienta para llevar a cabo análisis fiables de hipótesis urbanísticas y en lo que respecta al transporte y para planificar posibles situaciones.

El análisis del transporte brinda a los responsables de la planificación la posibilidad de:

- establecer estrategias y soluciones de transporte avanzadas y diseñadas para el futuro.
- realizar un análisis del tráfico.
- establecer la planificación de los servicios de transporte público.

• ¿Qué métodos se emplean para la modelización del transporte?

Durante la actualidad, se ha utilizado principalmente una metodología adicional conocida como el «modelo de las cuatro etapas», o el «modelo basado en viajes».

En la actualidad, en diversos lugares se han implementado métodos desagregados más detallados, conocidos como «modelos basados en actividades» o «modelos basados en agentes» (MBA), lo cual es nuestro caso. A continuación, se detallan ambos métodos:

A. Procesamiento de modelización tradicional de la demanda de movilidad en cuatro fases:

El procedimiento de las cuatro etapas se basa en una metodología concebida para la modelización de la demanda de movilidad urbana, regional y nacional. El modelo de

transporte de planificación agregada contiene cuatro etapas relacionadas con las opciones de viaje.

1. Generación de viajes: ¿cuáles son los números de viajes disponibles?
2. Distribución de viajes: ¿cuál es el lugar de destino en los viajes?
3. Selección de la forma de transporte: ¿qué forma de transporte se utiliza en cada viaje?
4. Revisión: ¿cuál es la ruta utilizada en cada trayecto?

B. Modelos desagregados basados en actividades / basados en agentes (MBA):

Dado que los modelos agregados se fundamentan en viajes, los modelos desagregados modelizan personas u hogares individualmente (es decir, agentes), con ubicaciones más precisas de los hogares y las actividades (coordenadas cartesianas exactas, en nuestro caso).

Si se deseaba representar a una población total (el cual no es nuestro caso), ya que no se dispone de datos individuales de toda la población, se generarán unos «datos demográficos artificiales» de los hogares y las personas a través de datos estadísticos.

La comprensión del tiempo y el espacio también suele ser más precisa, de modo que la estimación de los desplazamientos será más precisa.

Los MBA generan planes de la actividad diaria que cubren las actividades relevantes de cada persona, junto con su ubicación, horario, modo de transporte y ruta. Este registro de viaje artificial brinda un nivel de detalle espacial y temporal significativo en los resultados del modelo con el fin de supervisarlos. Dado que los resultados se generan como recorridos, viajes y actividades individuales, son más sencillos de comprender, ya que es todo muy visual e interactivo.

En conclusión, la modelización a través de agentes representa un progreso significativo en la planificación del transporte y el análisis de la movilidad urbana. Se brinda a los planificadores y diseñadores de políticas la oportunidad de evaluar de manera más eficiente el impacto de las intervenciones en transporte, asegurando que las soluciones planteadas se ajusten de manera cercana a las necesidades reales de los usuarios. Es esencial para elaborar sistemas de transporte más eficaces, seguros y sostenibles [2].

3.2 MATSIM

- **Descripción:**

MATSim (Multi-Agent Transport Simulation) es una plataforma de modelización y simulación de transporte que se fundamenta en el enfoque de sistemas multi-agente. Esta herramienta se emplea para simular y analizar políticas de transporte y movilidad a escala global, brindando un marco para la modelización del comportamiento de los viajes y las respuestas a las políticas en entornos urbanos y regionales [3].

- **Características:**

- **Modelización basada en agentes:** MATSim caracteriza a cada individuo como un agente independiente con un plan de actividades y viajes diarios personalizados. Cada persona toma decisiones en función de sus necesidades y el entorno, lo cual posibilita una simulación dinámica y realista del comportamiento de viaje en entornos urbanos y regionales.

- **Simulación de actividades diarias:** Los agentes en MATSim no solo transitan desde un punto A hasta un punto B, sino que llevan a cabo una serie de tareas definidas (tales como trabajar, comprar, estudiar), cada una con una ubicación y tiempo asignados. Esto brinda la oportunidad de modelar el impacto de las actividades diarias en los patrones de transporte y, de igual modo, viceversa.

- **Interacciones de Red y Ruta:** MATSim compuesto por complejas redes de transporte y algoritmos de ruta con el objetivo de simular con nitidez los flujos de tráfico. Los agentes seleccionan rutas que se fundamentan en la rapidez y conveniencia, evidenciando la capacidad de competir por el espacio de la carretera y otros recursos de transporte.

- **Gestión de grandes conjuntos de información:** MATSim abarca eficazmente grandes conjuntos de datos geográficos y de transporte, lo cual posibilita la visualización de áreas metropolitanas exhaustivas.

- **Visualización y el análisis:** Integración con herramientas como SIMUNTO VIA para la visualización avanzada de los resultados de la simulación, lo cual posibilita la interpretación y presentación de datos complejos [4].

- **Aplicación en el Contexto de Este Proyecto:**

En este proyecto, MATSim se ha empleado con el propósito de elaborar un modelo detallado de la movilidad en Valencia y su zona metropolitana, enfocado principalmente en el tratamiento de datos.

La capacidad de MATSim para evaluar el comportamiento de transporte basado en agentes ha sido esencial para:

- Poder comprender de manera más detallada la movilidad y el transporte, así como la comprensión del concepto "agente" y su funcionamiento.
- Sentar las bases para un futuro estudio enfocado a cualquier tipo de necesidad. Se pueden llevar a cabo estudios como optimizar la red de transporte, desarrollar nuevas estrategias de transporte, así como evaluar las políticas de movilidad.

En resumen, MATSim brinda una plataforma exhaustiva y flexible para la simulación detallada del transporte y la movilidad urbana, lo cual la convierte en una herramienta esencial para la planificación y evaluación de políticas en el ámbito del transporte.

4. METODOLOGÍA

4.1 CREACIÓN DEL ENTORNO DE DESARROLLO

- **Instalación de Herramientas Necesarias:** Descripción de la instalación y configuración de Java, Maven, y otras herramientas de software requeridas:

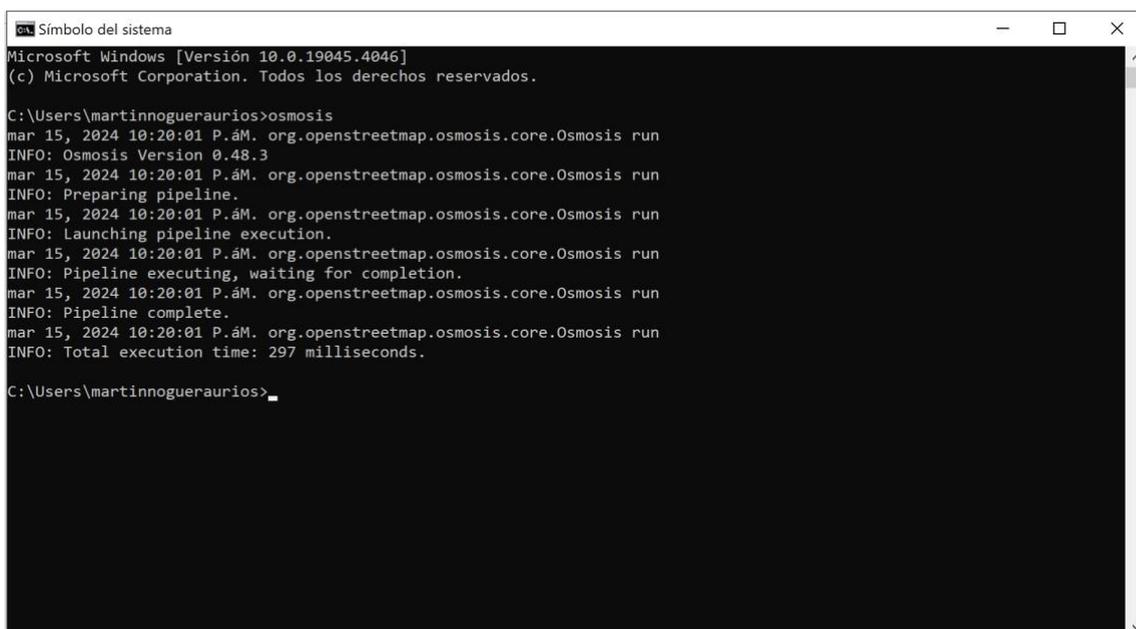
Para llevar a cabo este proyecto, lo primordial es instalar una serie de herramientas necesarias como Java JDK y Maven entre muchas otras, para así construir las bases sobre las que nuestro programa funcionará.

Lo primero es instalar java JDK 17.0.2 y apache Maven – 3.9.2 y agregar el path de Java JDK (Java Development Kit) y de Maven en las variables del sistema, específicamente dentro de las variables de entorno **PATH**, permitiendo así que el sistema operativo reconozca los comandos de Java y Maven desde cualquier directorio en la línea de comandos. Lo cual significa que puedo ejecutar comandos como **java**, **javac**, o **mvn** desde cualquier lugar sin necesidad de especificar el path completo al ejecutable.

Sus funciones serán explicadas en el siguiente apartado.

También se lleva a cabo la instalación de osmosis 0.48.3 y su posterior agregado a las variables de entorno PATH, para poder usar los comandos de osmosis con libertad, los cuales se tratarán con más profundidad en 4.2.1.

Al instalar osmosis 0.48.3, se introduce el comando de activación “osmosis” en el cmd para así poder hacer uso de las numerosas y útiles funciones que depara.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4046]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\martinnogueraurios>osmosis
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version 0.48.3
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline.
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution.
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion.
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline complete.
mar 15, 2024 10:20:01 P.ám. org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 297 milliseconds.

C:\Users\martinnogueraurios>
```

Ilustración 2. Activación osmosis

Se explicarán sus funciones en el siguiente apartado.

Otra herramienta muy importante que se ha utilizado, es JOSM, la cual instalándola, ha servido para poner unir distintas capas o archivos .osm en uno solo, además de también convertirlo a archivo.xml debido a la incorporación de plugins, todo esto se verá con más profundidad en 4.2.1

- **Configuración de GitHub:** GitHub ayudará a crear el repositorio a partir del cual construiremos el proyecto:

En primer lugar, se lleva a cabo la instalación de git desde internet, para posteriormente poder clonar localmente dicho repositorio sobre el que trabajaremos (para ello se necesitará su URL) [5].

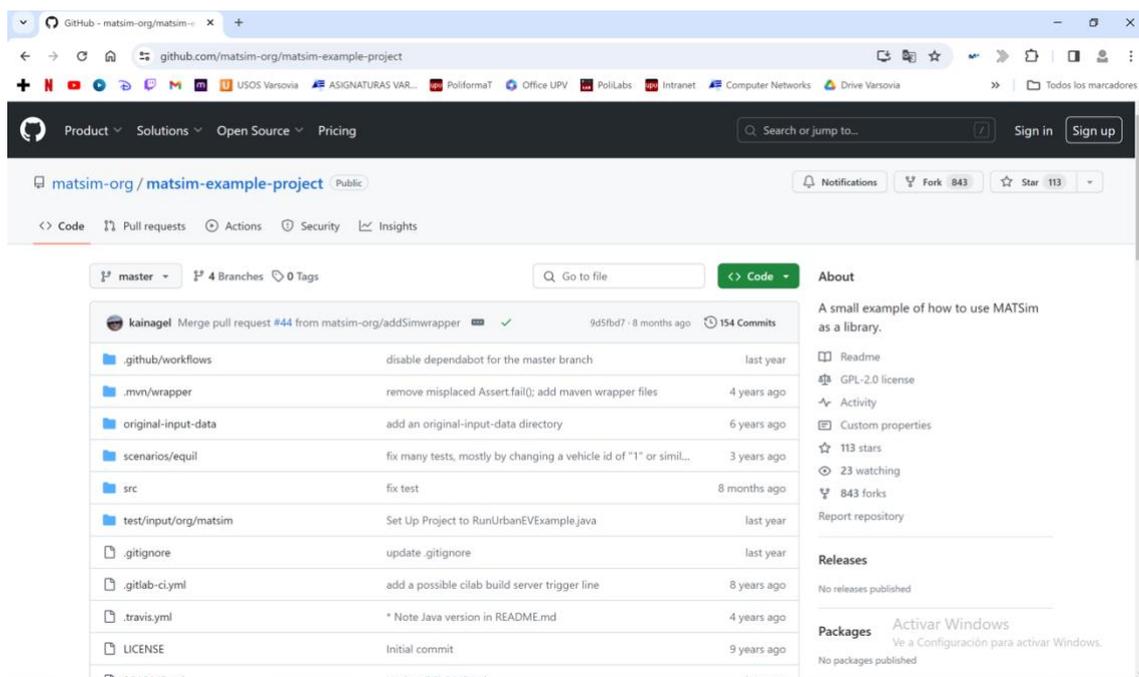


Ilustración 3. Repositorio base de GitHub

Para clonar el repositorio, hay que dirigirse al directorio donde se desea clonarlo y hacer click derecho “git bash here” donde se escribirá el comando “git init” para así poder inicializar el repositorio y después clonarlo en dicha ubicación:

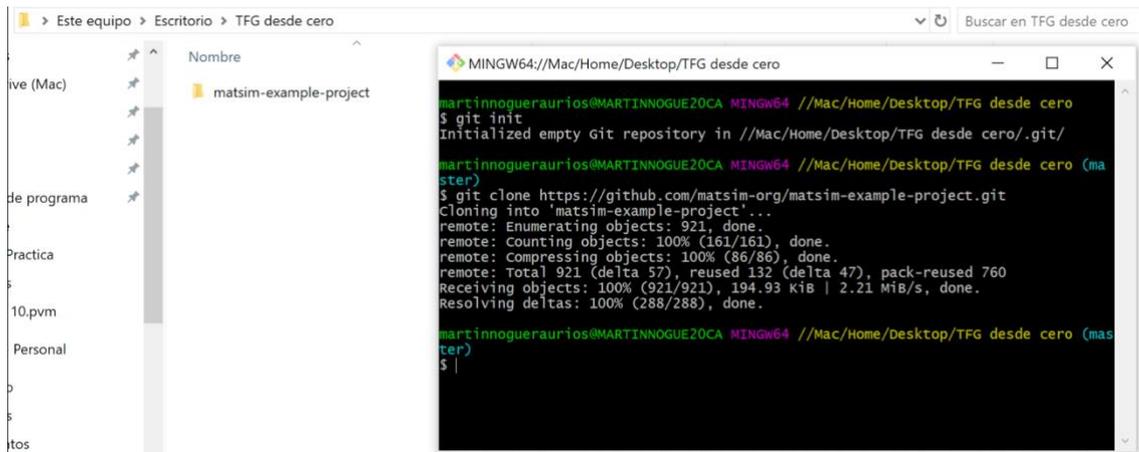


Ilustración 4. Clonación local del repositorio

Tras ejecutar el comando “git init”, se ejecuta “git clone”, cuya estructura es “git clone URL del repositorio de GitHub”.

De esta forma, ya se tiene el repositorio de GitHub clonado en la ubicación deseada, listo para poder utilizarlo y trabajar con él.

- **Preparación del Entorno Local:**

Todos los programas y herramientas mencionados anteriormente (Java JDK, Apache Maven, Osmosis, JOSM, GitHub) contribuyen al desarrollo y la gestión del código del entorno local.

1. Java JDK es el entorno de desarrollo de Java que proporciona las herramientas necesarias para desarrollar aplicaciones Java, incluyendo el compilador Java (javac) y la máquina virtual Java (JVM). Java JDK necesita estar instalado en la computadora local para compilar y ejecutar las aplicaciones Java. Es fundamental para ejecutar MATSim, que está desarrollado en Java [6].

2. Apache Maven es una herramienta que maneja dependencias de proyectos y facilita la construcción y empaquetado de proyectos Java. Maven se instala localmente para gestionar las bibliotecas y dependencias del proyecto, asegurando que todas las librerías necesarias estén disponibles y sean coherentes en el entorno de desarrollo [7].

3. Osmosis es una herramienta de línea de comandos para trabajar con datos OpenStreetMap (OSM), utilizada para filtrar o convertir grandes conjuntos de datos geográficos. Se ejecuta localmente para procesar y preparar datos geográficos que luego se usan en las simulaciones de MATSim [8].

4. JOSM es un editor para OpenStreetMap que permite la manipulación de los datos OSM. Se instala en el sistema local para editar manualmente los mapas y datos que serán utilizados en MATSim [9].

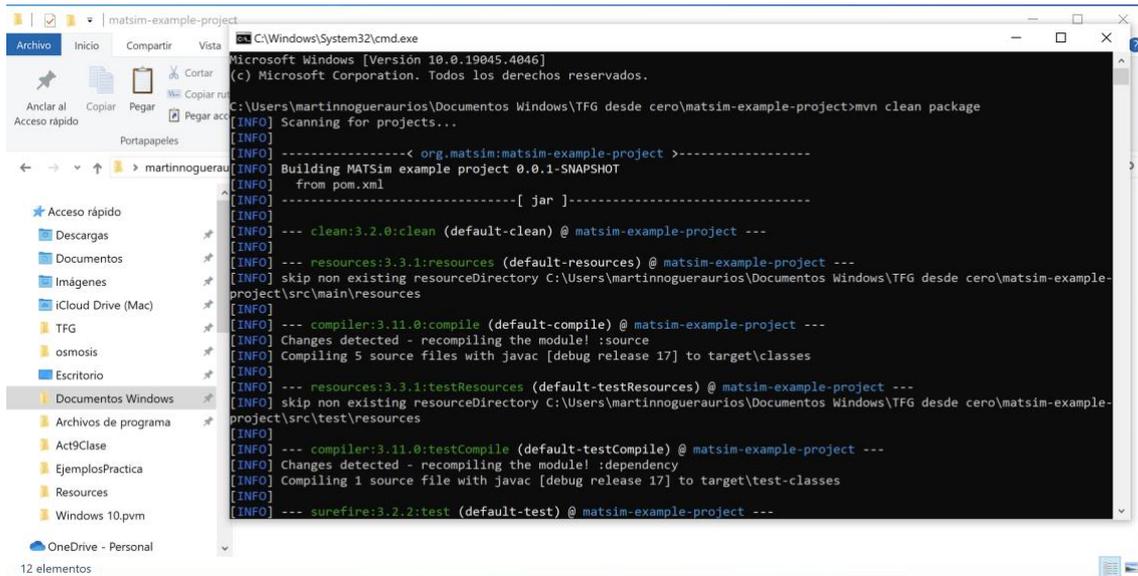
5. GitHub es una plataforma para la colaboración de código, que utiliza Git. Aunque GitHub es un servicio basado en la nube, la herramienta Git se instala localmente para gestionar el repositorio de código manualmente.

6. SNAPSHOT

También cabe nombrar el uso del ejecutable **SNAPSHOT.jar**, el cual contribuye a la creación del entorno de simulación.

Una vez se tiene el proyecto (el repositorio de GitHub clonado en la ubicación deseada), se creará el archivo ejecutable el cual se va a encargar de la simulación del escenario (matsim-example-project-0.0.1-SNAPSHOT). Para ello, se necesitará ejecutar en la carpeta del proyecto el siguiente código CMD:

mvn clean package:



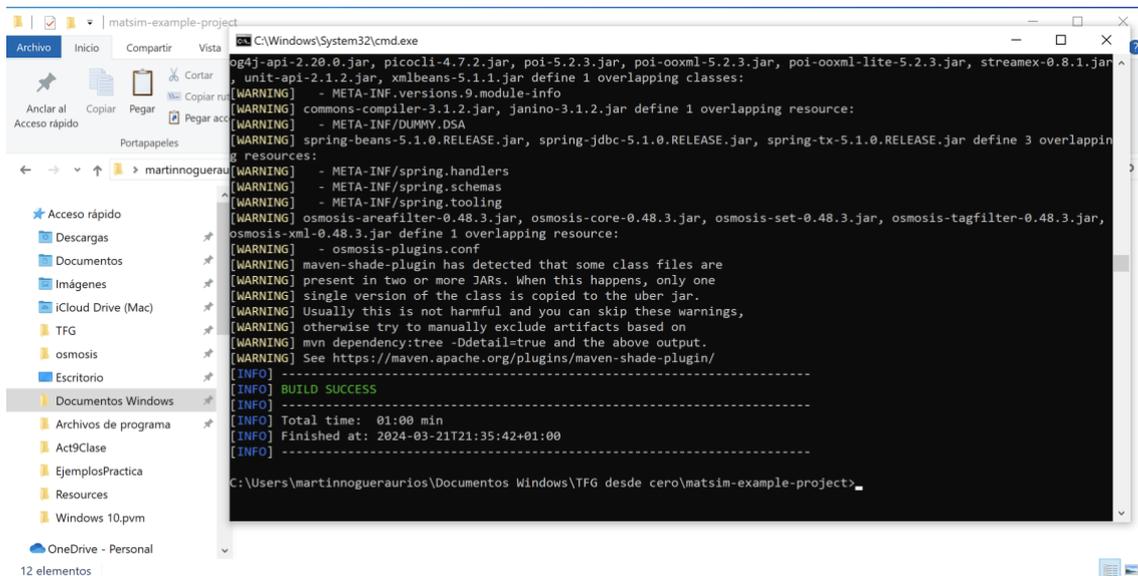
```

Microsoft Windows [Versión 10.0.19045.4046]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\martinnogueraurios\Documents\Windows\TFG desde cero\matsim-example-project>mvn clean package
[INFO] Scanning for projects...
[INFO]
-----< org.matsim:matsim-example-project >-----
[INFO] Building MATSim example project 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO]
-----[ jar ]-----
[INFO]
--- clean:3.2.0:clean (default-clean) @ matsim-example-project ---
[INFO]
--- resources:3.3.1:resources (default-resources) @ matsim-example-project ---
[INFO] skip non existing resourceDirectory C:\Users\martinnogueraurios\Documents\Windows\TFG desde cero\matsim-example-project\src\main\resources
[INFO]
--- compiler:3.11.0:compile (default-compile) @ matsim-example-project ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 5 source files with javac [debug release 17] to target\classes
[INFO]
--- resources:3.3.1:testResources (default-testResources) @ matsim-example-project ---
[INFO] skip non existing resourceDirectory C:\Users\martinnogueraurios\Documents\Windows\TFG desde cero\matsim-example-project\src\test\resources
[INFO]
--- compiler:3.11.0:testCompile (default-testCompile) @ matsim-example-project ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 1 source file with javac [debug release 17] to target\test-classes
[INFO]
--- surefire:3.2.2:test (default-test) @ matsim-example-project ---

```

Ilustración 5. Instalación ejecutable simulación (1/2)



```

og4j-api-2.20.0.jar, picocli-4.7.2.jar, poi-5.2.3.jar, poi-ooxml-5.2.3.jar, poi-ooxml-lite-5.2.3.jar, streamex-0.8.1.jar,
unit-api-2.1.2.jar, xmlbeans-5.1.1.jar define 1 overlapping classes:
[WARNING] - META-INF/versions.9/module-info
[WARNING] commons-compiler-3.1.2.jar, janino-3.1.2.jar define 1 overlapping resource:
[WARNING] - META-INF/DUMMY.DSA
[WARNING] spring-beans-5.1.0.RELEASE.jar, spring-jdbc-5.1.0.RELEASE.jar, spring-tx-5.1.0.RELEASE.jar define 3 overlapping
resources:
[WARNING] - META-INF/spring.handlers
[WARNING] - META-INF/spring.schemas
[WARNING] - META-INF/spring.tooling
[WARNING] osmosis-areafilter-0.48.3.jar, osmosis-core-0.48.3.jar, osmosis-set-0.48.3.jar, osmosis-tagfilter-0.48.3.jar,
osmosis-xml-0.48.3.jar define 1 overlapping resource:
[WARNING] - osmosis-plugins.conf
[WARNING] maven-shade-plugin has detected that some class files are
[WARNING] present in two or more JARs. When this happens, only one
[WARNING] single version of the class is copied to the uber jar.
[WARNING] Usually this is not harmful and you can skip these warnings,
[WARNING] otherwise try to manually exclude artifacts based on
[WARNING] mvn dependency:tree -Ddetail=true and the above output.
[WARNING] See https://maven.apache.org/plugins/maven-shade-plugin/
[INFO]
-----
[INFO] BUILD SUCCESS
[INFO]
-----
[INFO] Total time: 01:00 min
[INFO] Finished at: 2024-03-21T21:35:42+01:00
[INFO]
-----
C:\Users\martinnogueraurios\Documents\Windows\TFG desde cero\matsim-example-project>

```

Ilustración 6. Instalación ejecutable simulación (2/2)

De esta manera, ya se tiene el archivo ejecutable creado en la ubicación deseada:

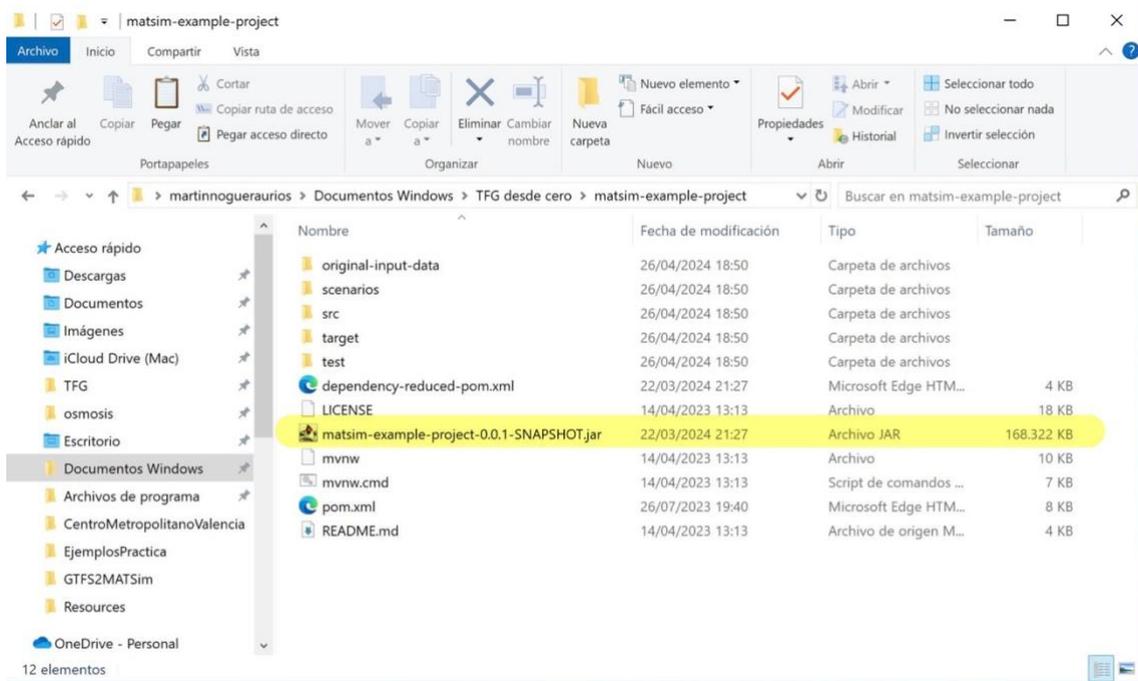


Ilustración 7. Ejecutable instalado

Dicho archivo JAR pertenece a un proyecto de MATSim “de ejemplo”, a partir del cual se construirá el proyecto adaptándolo a las respectivas necesidades.

Para la simulación de nuestro proyecto con SIMUNTO VIA se abrirá el archivo .jar obtenido anteriormente y se abrirá un display básico en el cual habrá que seleccionar el archivo config.xml que se quiere simular y la carpeta donde se generarán todos los archivos tras la simulación; solo será necesario incluir este archivo ya que en su interior se especifica todo lo demás (qué network.xml y plans.xml se utilizan, las diferentes configuraciones de cada actividad etc.) . De todos estos archivos se hablará con más detalle en 4.3.

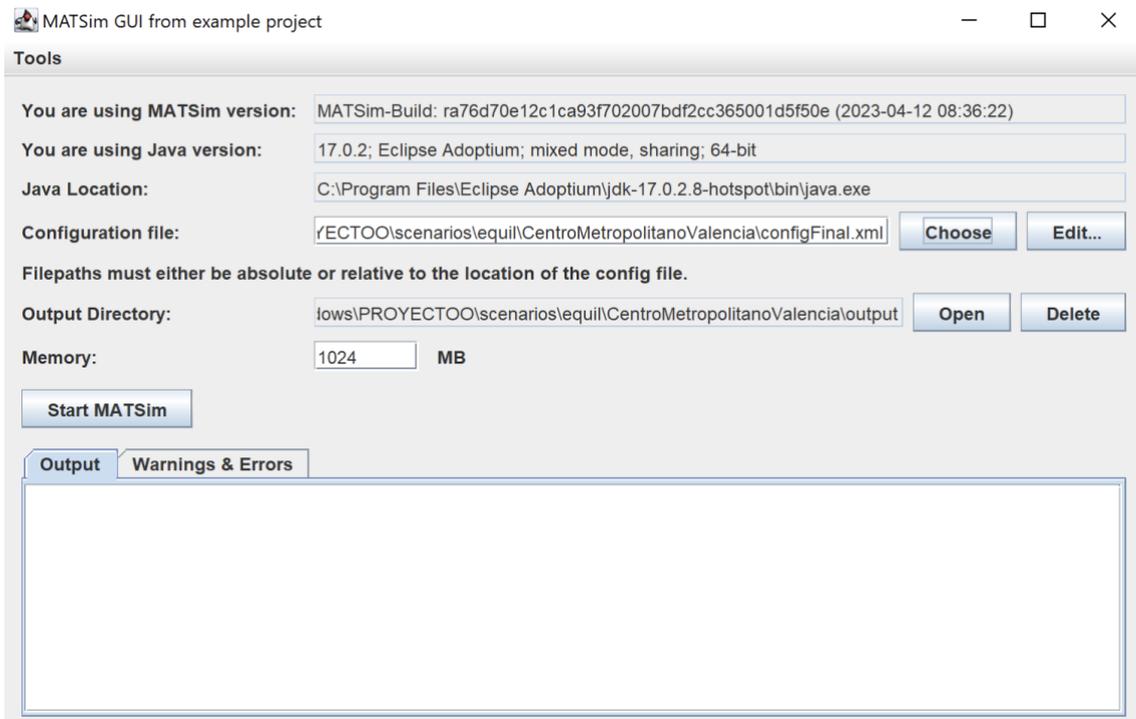


Ilustración 8. Display ejecutable

Si observamos, se incluye el archivo de entrada configFinal.xml, y una vez ejecutado sin errores, los archivos de salida se ubicarán en la carpeta “output”, tal como se indica en output directory:

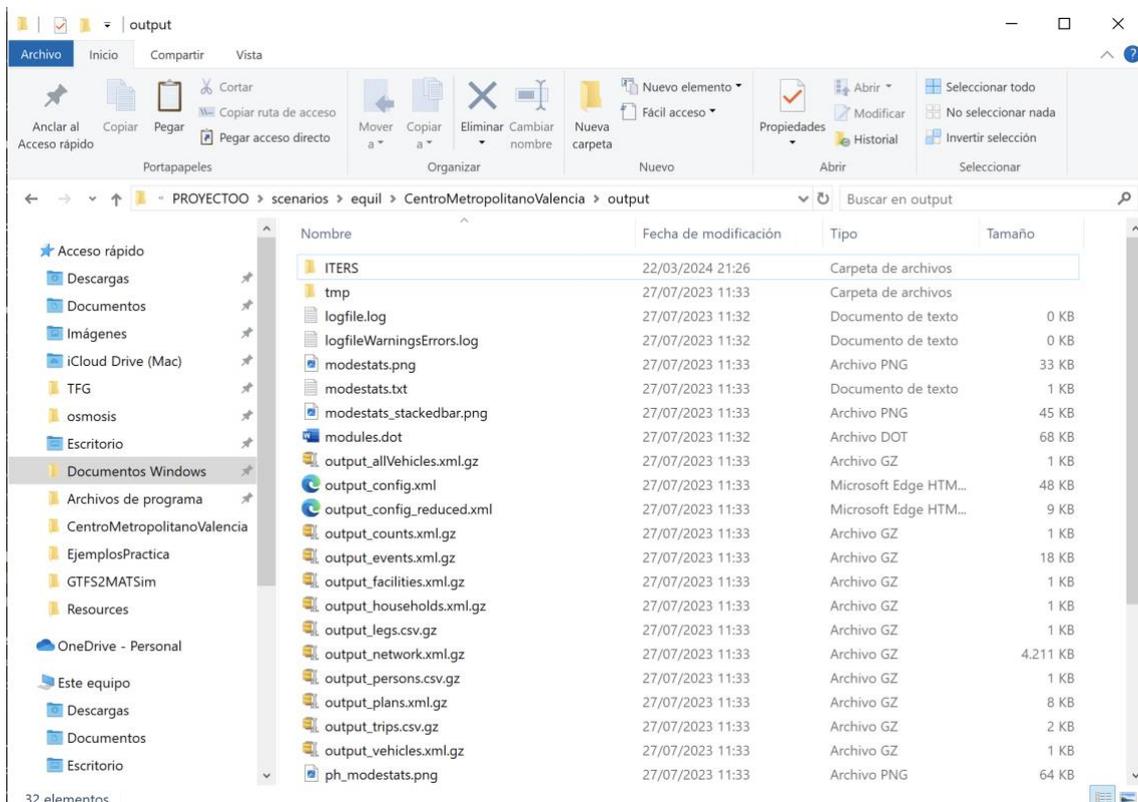


Ilustración 9. Directorio archivos generados

Todo esto permite, además, ir viendo los resultados ante los cambios y mejoras que se van implementando.

7. Visual Studio Code

Una de las partes más importantes de este entorno local, es el IDE (entorno de desarrollo integrado) utilizado, el cual es **VISUAL STUDIO CODE** donde se ha llevado a cabo la creación y modificación de todas las scripts o archivos xml utilizados para la simulación, principalmente el archivo config.xml, entre otros.

Una vez instalado VSCode, se instalan extensiones de Java y Maven entre otras para su correcto funcionamiento:

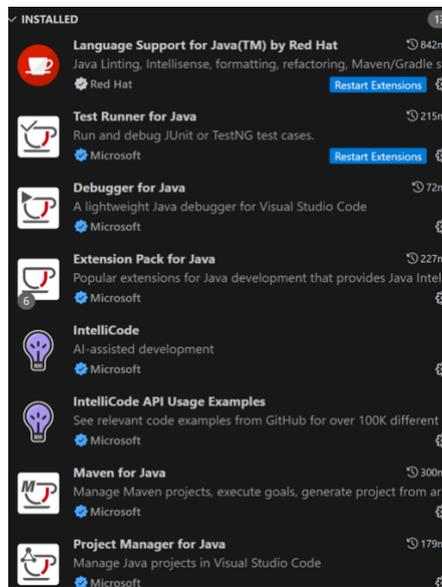


Ilustración 10. Extensiones VSCode (1/2)

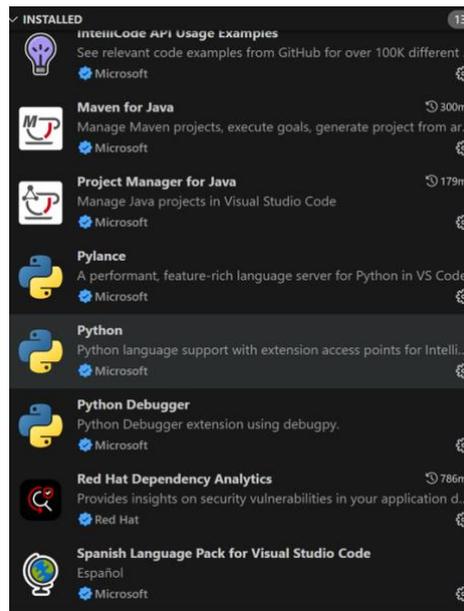


Ilustración 11. Extensiones VSCode (2/2)

Todas estas herramientas y programas, junto con el software complementario utilizado llamado SIMUNTO VIA, forman el Entorno Local en su totalidad.

Sobre el software **Simunto Via** se hablará detenidamente en el punto 4.4, ya que se trata del programa donde se lleva a cabo toda la simulación y la visualización de los archivos .xml, además de contribuir a averiguar si las scripts están correctamente creadas o no.

Con todo esto correctamente instalado, tenemos todas las herramientas disponibles para llevar a cabo el proyecto.

4.2 RECOPIACIÓN, TRATAMIENTO Y CONVERSIÓN DE DATOS PARA EL MODELADO EN MATSIM

Existen diferencias significativas entre recopilación de datos y conversión de datos. Estos términos describen etapas diferentes y esenciales en el proceso de manejo de datos, especialmente en proyectos de modelización y análisis como los que involucran sistemas de simulación como MATSim:

- **Recopilación de Datos**

La recopilación de datos implica el proceso de obtener datos desde diversas fuentes (Internet) para ser utilizados en un análisis o proyecto específico. En el contexto de un estudio de movilidad urbana:

Datos recolectados directamente a través de encuestas, observaciones en campo, etc. Y datos recolectados a través de fuentes existentes como bases de datos gubernamentales, informes de tráfico, datos de GPS de servicios de transporte público, etc.

La recopilación de datos es el primer paso crucial y establece la base de información sobre la que se construirá el análisis.

- **Tratamiento de Datos**

El tratamiento de datos (o procesamiento de datos), en nuestro caso se reduce a:

- **Limpieza de Datos:** Eliminar o corregir datos erróneos, incompletos, duplicados o irrelevantes [10].

- **Conversión de Datos**

La conversión de datos se refiere específicamente al proceso de cambiar el formato de los datos de un formato a otro para que sean compatibles con diferentes herramientas o sistemas. En un proyecto de MATSim, por ejemplo, esto podría incluir:

- **Conversión de Formatos:** Transformar datos desde diferentes formatos, a formatos específicos requeridos por el software, como archivos XML o formatos específicos de MATSim.

Esta etapa es crítica para asegurar que los datos puedan ser efectivamente utilizados por las herramientas de análisis o simulación previstas [11].

Conclusión

Cada una de estas etapas es crucial y tiene su propio conjunto de técnicas y herramientas. La recopilación de datos asegura que se tengan los datos necesarios; el tratamiento de datos garantiza que los datos sean limpios y estructurados adecuadamente; y la conversión de datos asegura que los datos puedan ser utilizados por las aplicaciones o sistemas elegidos.

Una vez entendido esto, se procede a explicar todo el proceso desde cero, hasta la creación de la network:

• 4.2.1 Crear la Network sobre la que se va a trabajar

Primero se debe descargar de la web geofabrik, el mapa competo de la comunidad valenciana en su versión .osm.pbf, el cual es valencia-latest.osm.pbf [12] .

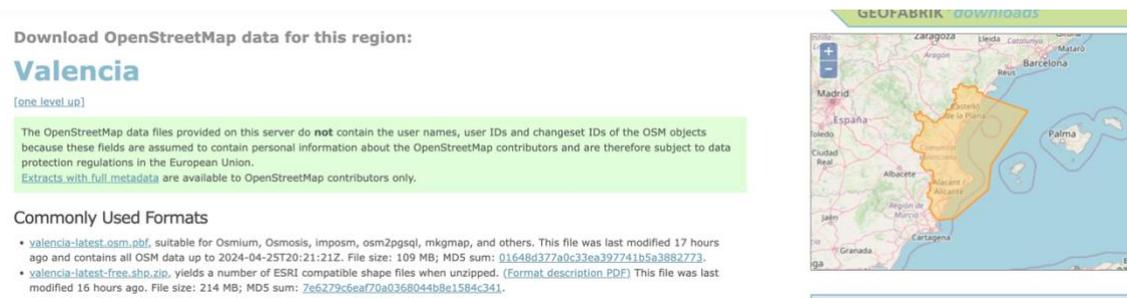


Ilustración 12. Página web geofabrik

Funciones osmosis (.pbf a .osm):

Este mapa descargado se utilizará para, a través de un comando de osmosis, recortar una a una las 4 capas de dicho mapa que posteriormente se unirán gracias a JOSM para poder crear así la network sobre la que se trabajará, la cual será el mapa de Valencia:

Para saber exactamente el área a delimitar, accedemos a la web de OpenStreetMap, y en el menú de recortar, se elige la opción de seleccionar manualmente un área diferente (no se recorta directamente el área deseada ya que excede el número de nodos permitido por OpenStreetMap). [13]

Esto sirve para saber exactamente las coordenadas norte, sur este y oeste en las que queremos delimitar cada una de las 4 capas (señaladas en amarillo):

Capa “centro”:

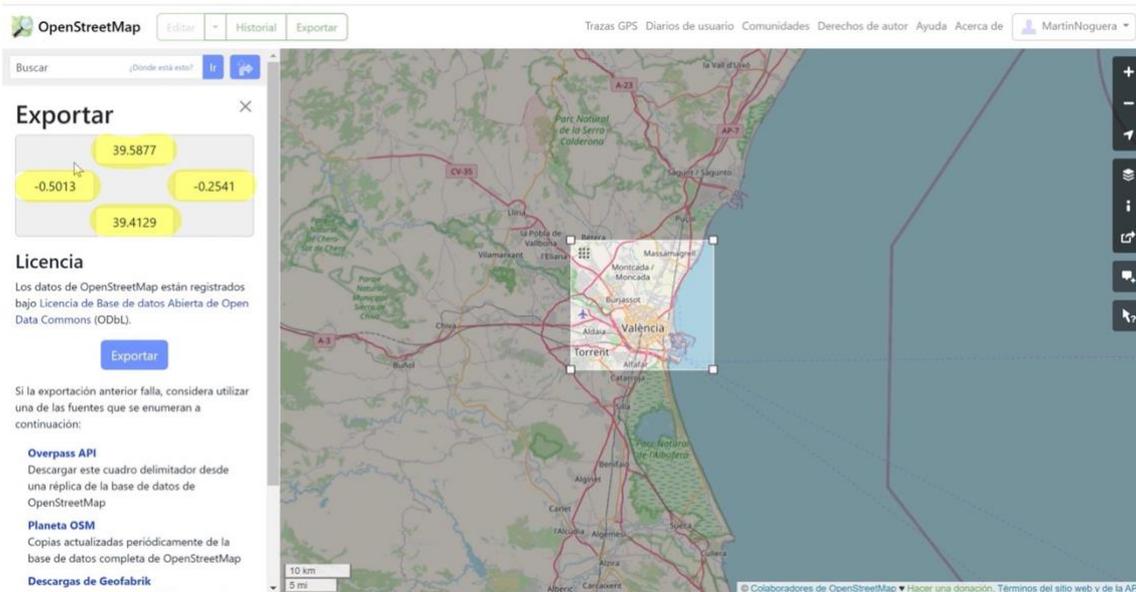


Ilustración 13. Coordenadas capa centro

Capa “norte”:

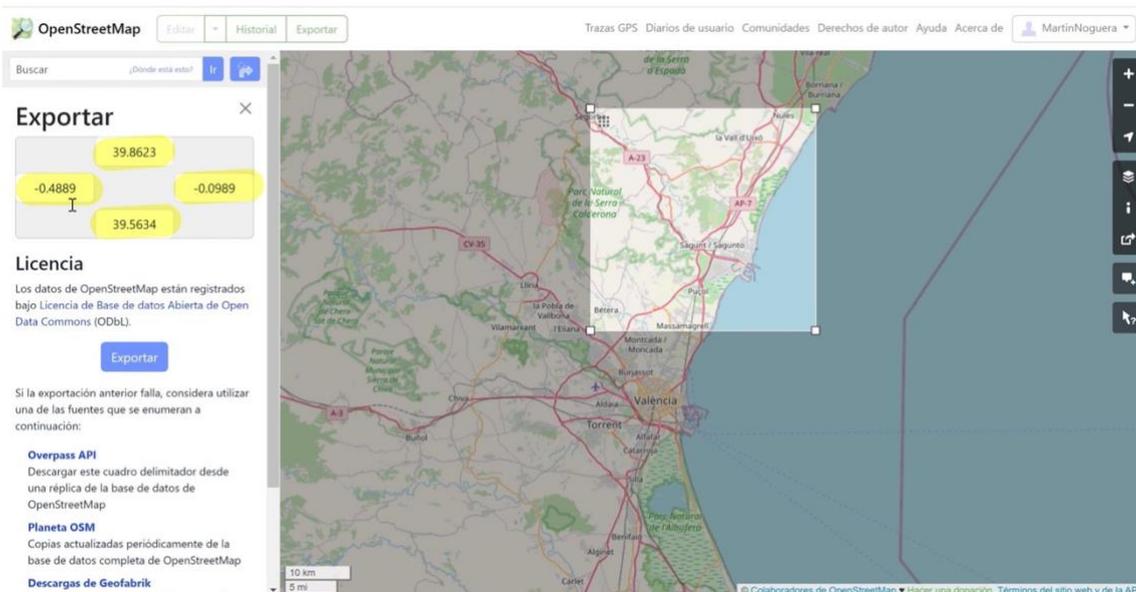


Ilustración 14. Coordenadas capa norte

Capa “sur”:

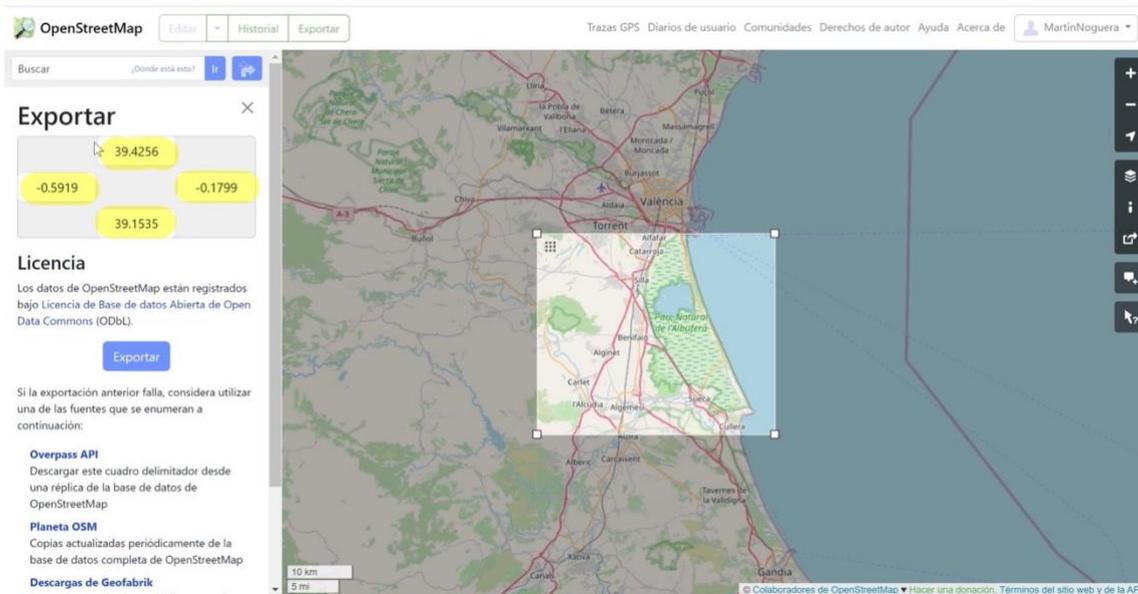


Ilustración 15. Coordenadas capa sur

Capa “Lliria”:

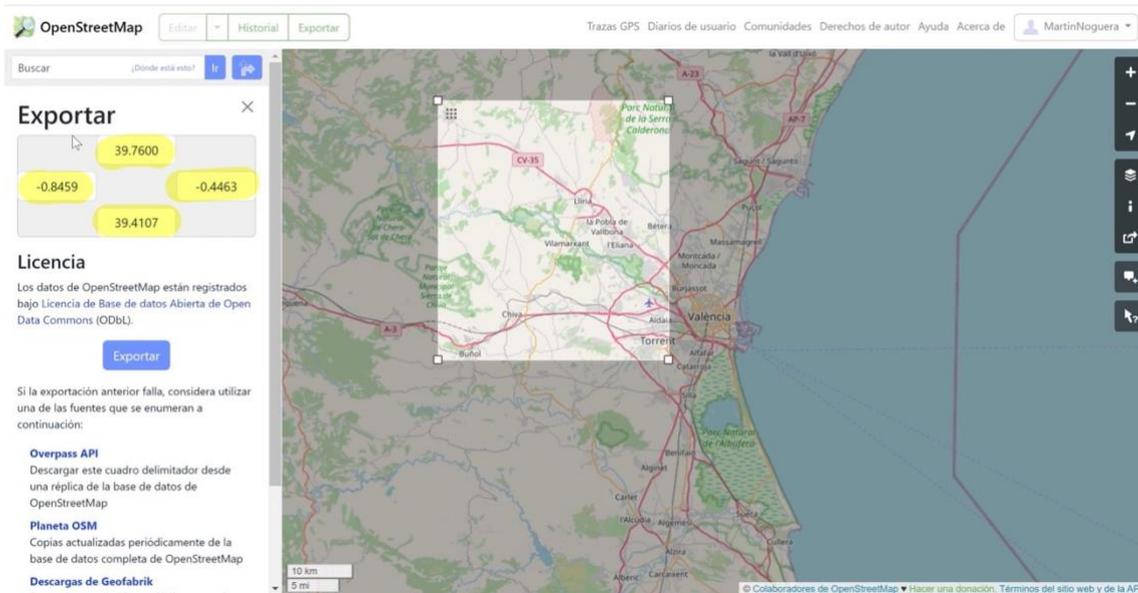


Ilustración 16. Coordenadas capa lliria

Ahora lo que se necesita es acotar el mapa descargado de geofabrik, el cual se ha renombrado como valenciainicial.pbf y así quedarse con la zona deseada. Para esto es necesario el uso de la herramienta Osmosis [14], la cual mediante el siguiente comando se acotarán dichas zonas.

En primer lugar, se sitúa el mapa (en formato .pbf) descargado en la misma ubicación que el programa osmosis:

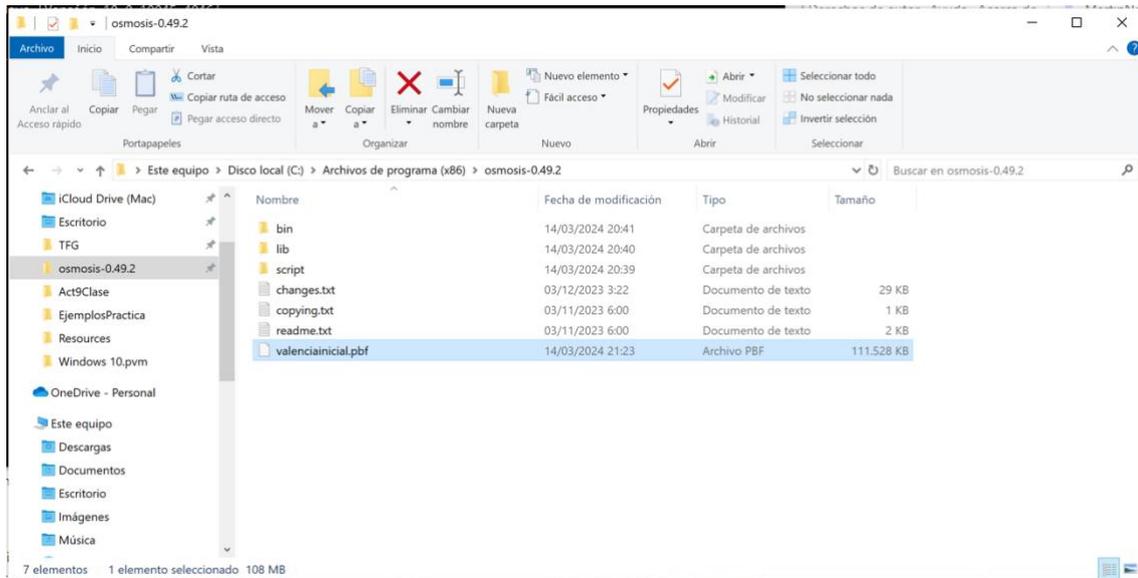


Ilustración 17. Ubicación mapa en formato .pbf

A continuación, se utiliza el siguiente comando de osmosis que sirve para recortar, anotando cada una de las coordenadas de las imágenes anteriores.

Dicho comando se ejecutará en el cmd en la ubicación donde están el mapa y el programa. Se usará el comando 4 veces:

Para obtener la capa “centro.osm”:

```
C:\Program Files (x86)\osmosis>osmosis --rb file=valenciainicial.pbf --bounding-box left=-0.5013 right=-0.2541 bottom=39.4129 top=39.5877 --write-xml file=centro.osm
mar 21, 2024 12:30:37 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version 0.48.3
mar 21, 2024 12:30:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline.
mar 21, 2024 12:30:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution.
mar 21, 2024 12:30:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion.
mar 21, 2024 12:30:50 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline complete.
mar 21, 2024 12:30:50 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 12766 milliseconds.
```

Ilustración 18. Recorte capa centro

Para obtener la capa “norte.osm”:

```
C:\Program Files (x86)\osmosis>osmosis --rb file=valenciainicial.pbf --bounding-box left=-0.4889 right=-0.0989 bottom=39.5634 top=39.8623 --write-xml file=norte.osm
mar 21, 2024 12:31:04 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version 0.48.3
mar 21, 2024 12:31:05 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline.
mar 21, 2024 12:31:05 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution.
mar 21, 2024 12:31:05 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion.
mar 21, 2024 12:31:13 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline complete.
mar 21, 2024 12:31:13 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 8669 milliseconds.
```

Ilustración 19. Recorte capa norte

Para obtener la capa “sur.osm”:

```
C:\Program Files (x86)\osmosis>osmosis --rb file=valenciainicial.pbf --bounding-box left=-0.5919 right=-0.1799 bottom=39.1535 top=39.4256 --write-xml file=sur.osm
mar 21, 2024 12:31:29 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version 0.48.3
mar 21, 2024 12:31:30 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline.
mar 21, 2024 12:31:30 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution.
mar 21, 2024 12:31:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion.
mar 21, 2024 12:31:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline complete.
mar 21, 2024 12:31:38 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 8758 milliseconds.
```

Ilustración 20. Recorte capa sur

Para obtener la capa “lliria.osm”:

```
C:\Program Files (x86)\osmosis>osmosis --rb file=valenciainicial.pbf --bounding-box left=-0.8459 right=-0.4463 bottom=39.4107 top=39.7600 --write-xml file=lliria.osm
mar 21, 2024 12:10:39 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version 0.48.3
mar 21, 2024 12:10:39 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline.
mar 21, 2024 12:10:40 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution.
mar 21, 2024 12:10:40 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion.
mar 21, 2024 12:10:49 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline complete.
mar 21, 2024 12:10:49 A.M. org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 9719 milliseconds.
```

Ilustración 21. Recorte capa lliria

De esta manera, no solo se consigue obtener las 4 capas usando comandos de osmosis, sino que también se realiza una conversión de archivo .pbf a archivo .osm

Se puede observar:

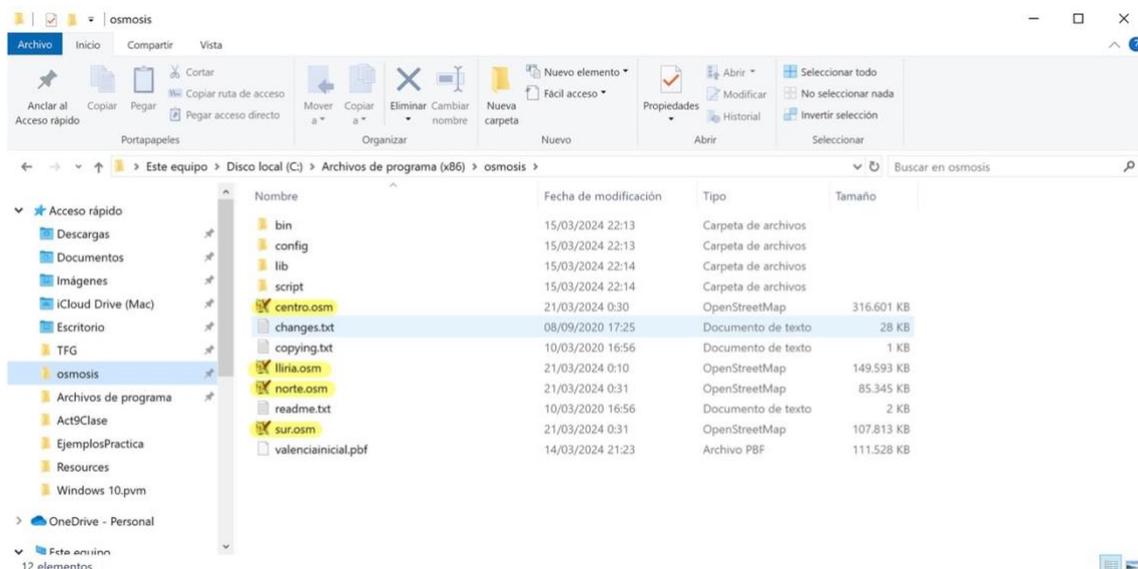


Ilustración 22.m Archivos recortados y en formato .osm

Llegados a este punto, se aprecia una clara recopilación de datos de geofabrik y openstreetmap.

Funciones JOSM (.osm a .xml):

JOSM, es un editor de archivos .osm, por lo que se abren en JOSM los 4 archivos generados con los comandos de osmosis:

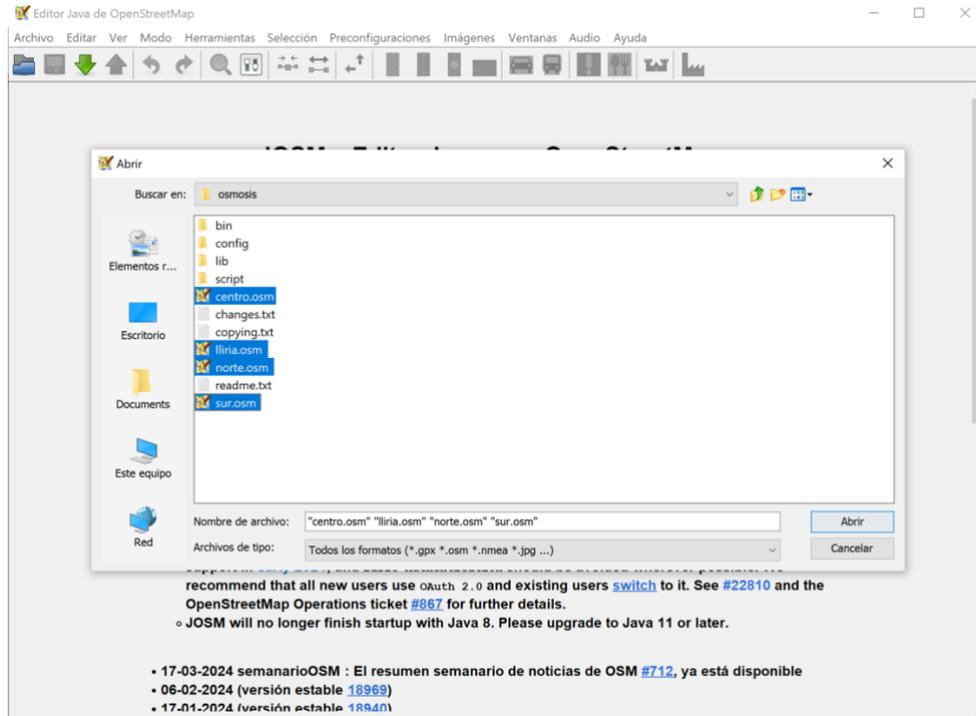


Ilustración 23. Archivos abiertos en JOSM

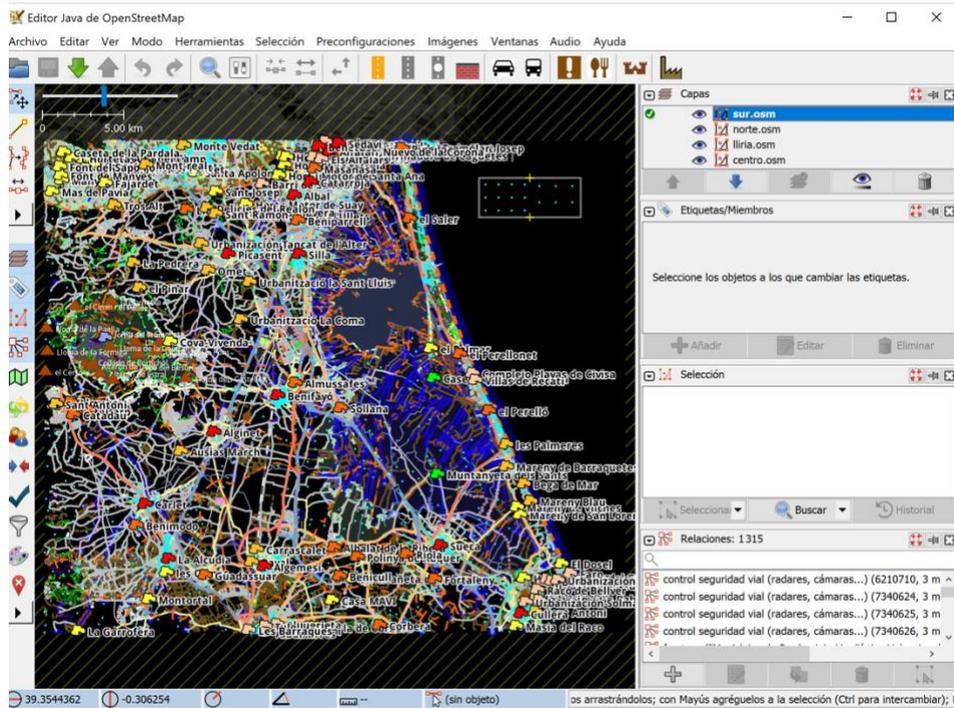


Ilustración 24. Aspecto archivos en JOSM

Una vez abiertos, se selecciona una de las capas abiertas, click derecho y se le da a la opción “unir” con otra de las 4 capas, y así respectivamente hasta que quede solo una capa (en .osm) que sea el resultado de la unión de las 4 originales.

Este sería el resultado de una primera unión entre la capa norte.osm y la capa sur.osm :

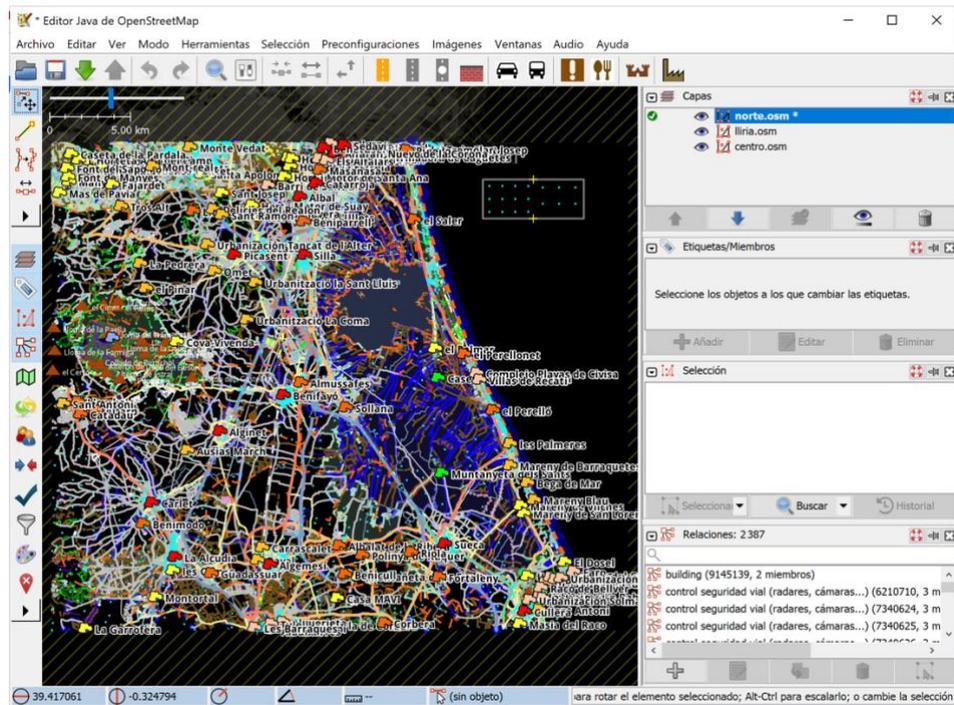


Ilustración 25. Unión de dos capas

Aquí se muestra un video uniendo las 4 capas:

https://drive.google.com/file/d/1025PCvBihNM5SSwptE8n_8hRsQQ7W8Y0/view?usp=sharing

Una vez que se tienen todas las capas unidas en un solo archivo.osm, se realiza una conversión a .xml para que la futura network sea compatible con MATSim.

Para ello, se tendrá que instalar el plugin de MATSim en JOSM:

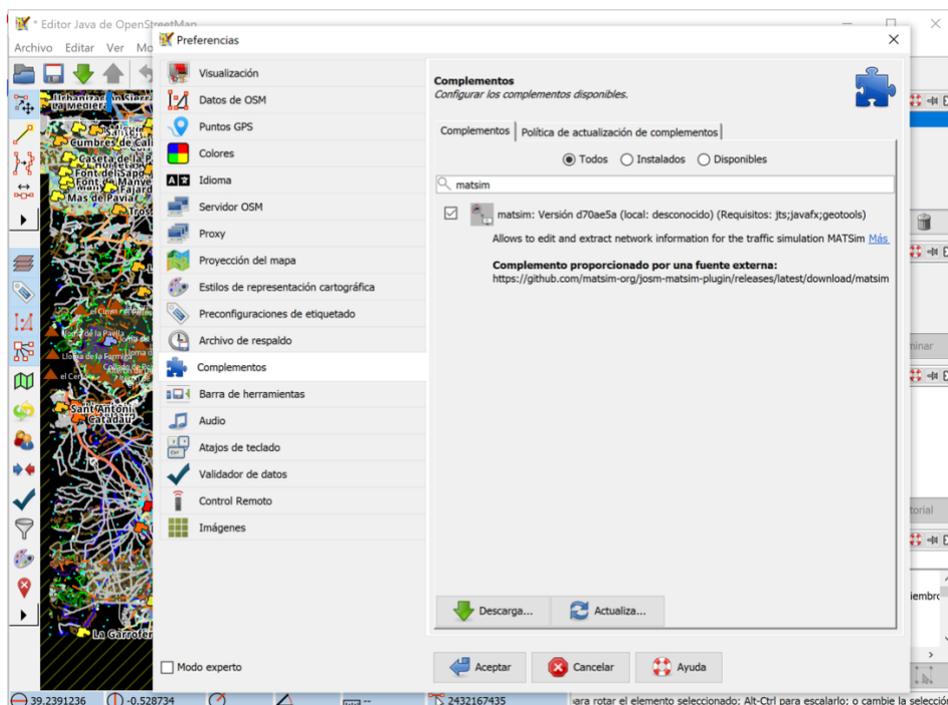


Ilustración 26. Plugin MATSim

Para posteriormente realizar dicha conversión (.osm a .xml):

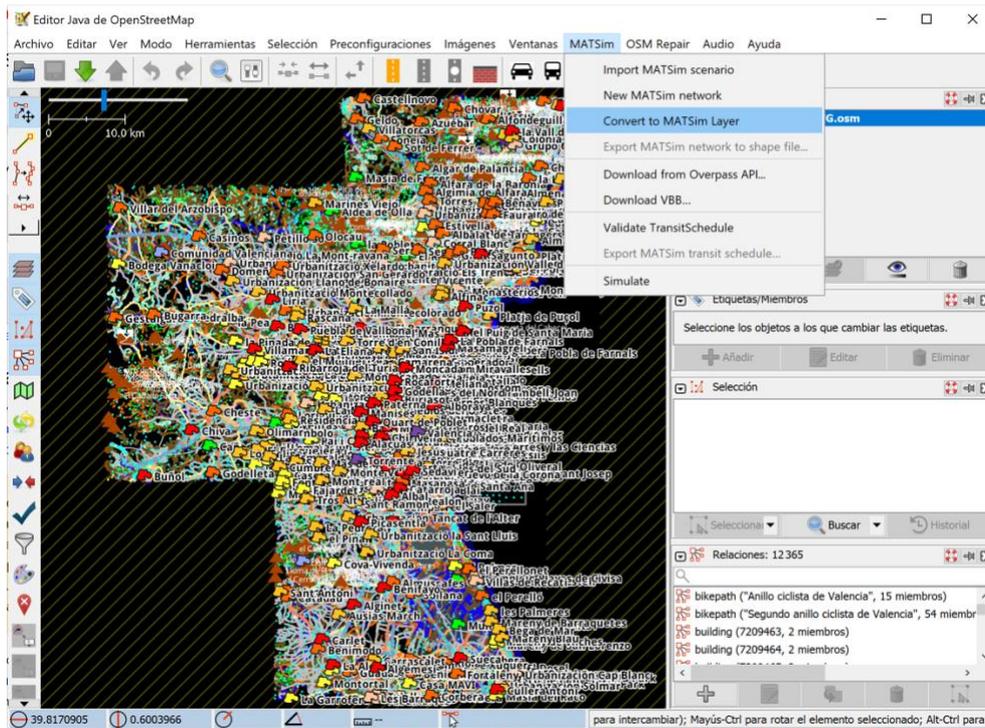


Ilustración 27. Conversión .osm a .xml

Y el archivo resultante de la conversión será la network.xml compatible con MATSim y con Simunto Via. Se puede observar su aspecto abriéndola en el software Simunto Via:

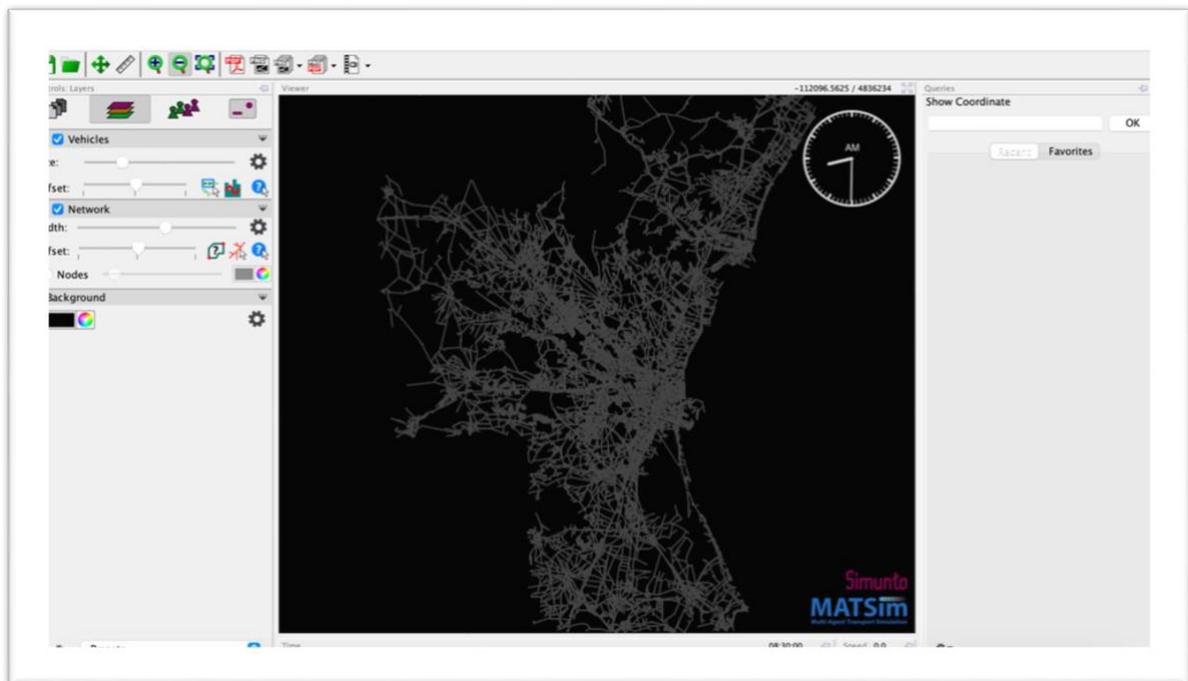


Ilustración 28. Aspecto network

El resultado es correcto, ya que es la unión de las 4 capas originalmente recortadas de geofabrik, por lo que se tiene ya la NETWORK sobre la que operar.

En conclusión, se puede observar cómo se han abordado todas las etapas de los datos:

Recopilación de datos:

- Al obtener el archivo valenciainicial.pbf de la web de geofabrik.

Tratamiento de datos:

- Al usar OpenStreetMap, se ha podido averiguar las coordenadas de cada una de las que serán las 4 capas que formarán la network.
- Situar el archivo valenciainicial.pbf en la ubicación adecuada (donde se encuentra el programa osmosis instalado)
- Con el editor JOSM, juntar las 4 capas o archivos .osm en una sola capa o archivo .osm

Conversión de datos:

- De .pbf a .osm: Utilizando los comandos osmosis para recortar el mapa inicial (en formato .pbf) y obtener 4 capas o archivos .osm
- De .osm a .xml: A través de instalar, en JOSM, un plugin de MATSim para poder convertir dicho archivo .osm (el resultante de juntar las 4 capas) en un “MATSim layer”, es decir, en un archivo .xml
- Dicho archivo .xml resultante, se trata de la NETWORK definitiva.

• 4.2.2 Obtención de config.xml

Por otra parte, como ya se ha nombrado anteriormente, una tarea indispensable es la clonación del repositorio de GitHub “matsim-example-project”, ya que dicho proyecto contiene una serie de archivos .xml predeterminados, el archivo config.xml en concreto; el cual sirve de base para todas las adiciones que vaya implementando hasta construir nuestro config definitivo, dicho archivo contiene toda la información

necesaria para la posterior simulación. Se explicará más detenidamente este archivo en el punto 4.3.2.

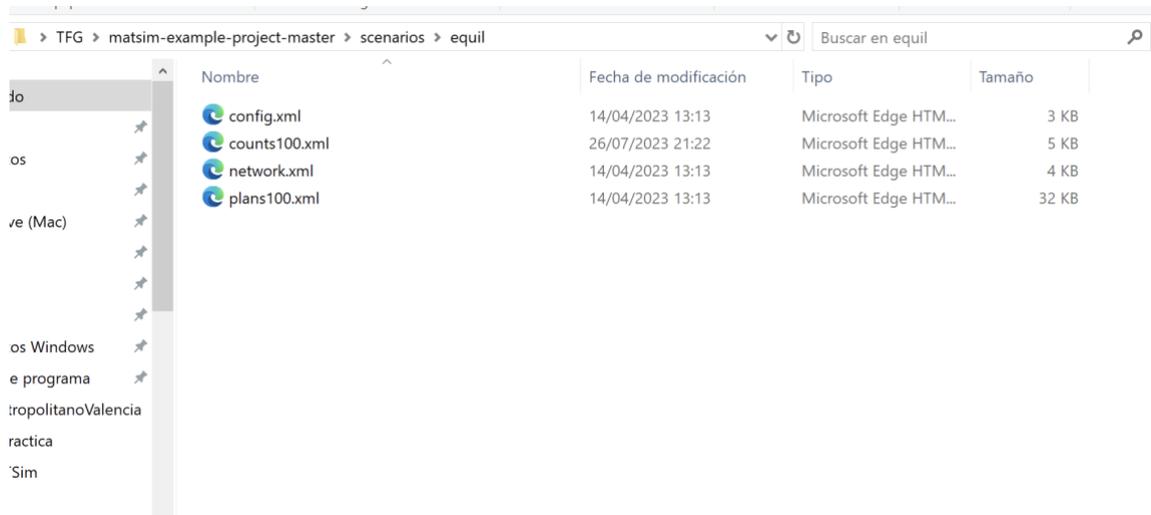


Ilustración 29. Contenido del repositorio inicial

```

1 <?xml version="1.0" ?>
2 <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
3 <config>
4
5   <module name="global">
6     <param name="randomSeed" value="4711" />
7     <param name="coordinateSystem" value="Atlantis" />
8   </module>
9
10  <module name="network">
11    <param name="inputNetworkFile" value="network.xml" />
12  </module>
13
14  <module name="plans">
15    <param name="inputPlansFile" value="plans100.xml" />
16  </module>
17
18  <module name="controller">
19    <param name="outputDirectory" value="./output" />
20    <param name="firstIteration" value="0" />
21    <param name="lastIteration" value="10" />
22  </module>
23
24  <module name="qsim">
25    <!-- "start/endTime" of MobSim (00:00:00 == take earliest activity time/ run as long as active vehicles exist) -->
26    <param name="startTime" value="00:00:00" />
27    <param name="endTime" value="00:00:00" />
28
29    <param name="snapshotperiod" value="00:00:00"/> <!-- 00:00:00 means NO snapshot writing -->
30  </module>
31
32  <module name="planCalcScore">
33    <param name="learningRate" value="1.0" />
34    <param name="BrainExpBeta" value="2.0" />
35
36    <param name="lateArrival" value="-18" />
37    <param name="earlyDeparture" value="-0" />
38    <param name="performing" value="+6" />
39    <param name="waiting" value="-0" />

```

Ilustración 30. Config predeterminado

En este proceso, hay una recopilación de datos evidente, al clonar el repositorio de GitHub de la web mencionada anteriormente. También tendrá lugar, posteriormente, el

tratamiento de datos al modificar dicho config para que se adecúe perfectamente a las necesidades sobre el proyecto.

- **4.2.3 Transporte público**

Por último, cabe hablar del **transporte público** y de cómo se van a incorporar dichos datos a nuestro proyecto. Ya que todo lo anterior está dedicado a todo tipo de agentes y vehículos, a excepción del transporte público (todo esto se verá con detalle más adelante).

En este proceso de incorporar las paradas de los autobuses urbanos e interurbanos de Valencia y su área metropolitana al proyecto, hay una clara recopilación de datos de la web del Gobierno de España, específicamente dos archivos zip referidos al autobús urbano e interurbano de Valencia [15] [16].

Posteriormente, se clonará el repositorio de GitHub llamado GTFS2MATSim en el lugar deseado (lo cual es también una clara recopilación de datos de un repositorio de GitHub): [17]

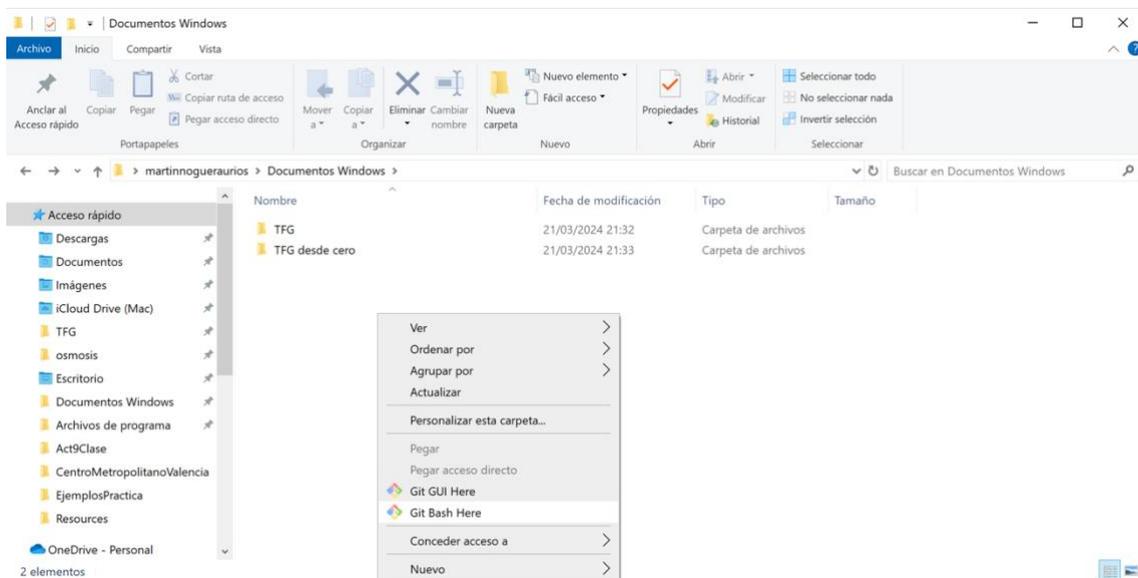


Ilustración 31. Clonación repositorio de GitHub GTFS2MATSim(1/3)

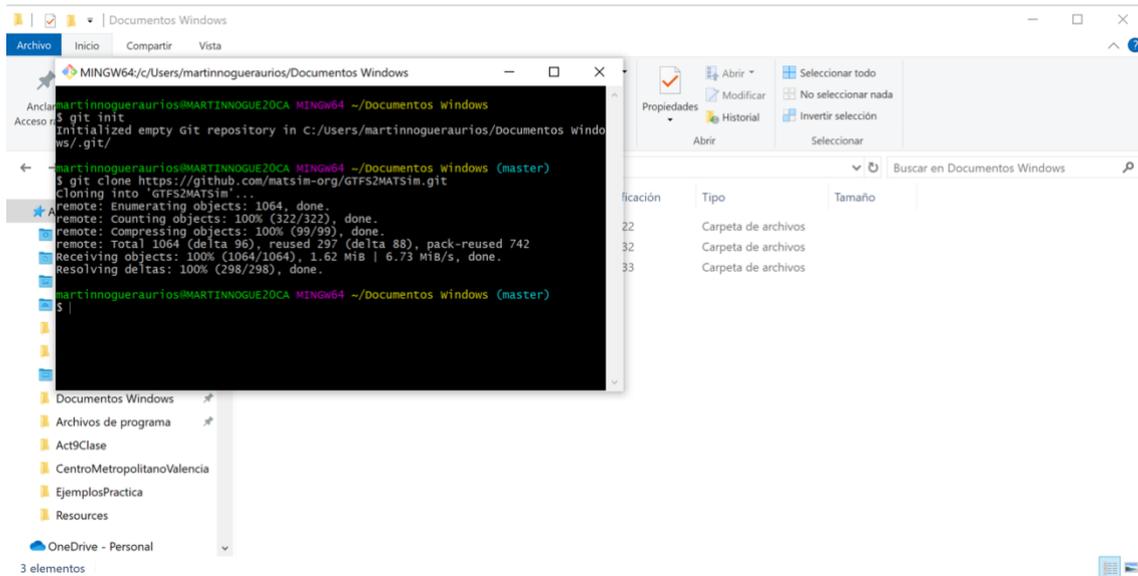


Ilustración 32. Clonación repositorio de GitHub GTFS2MATSim(2/3)

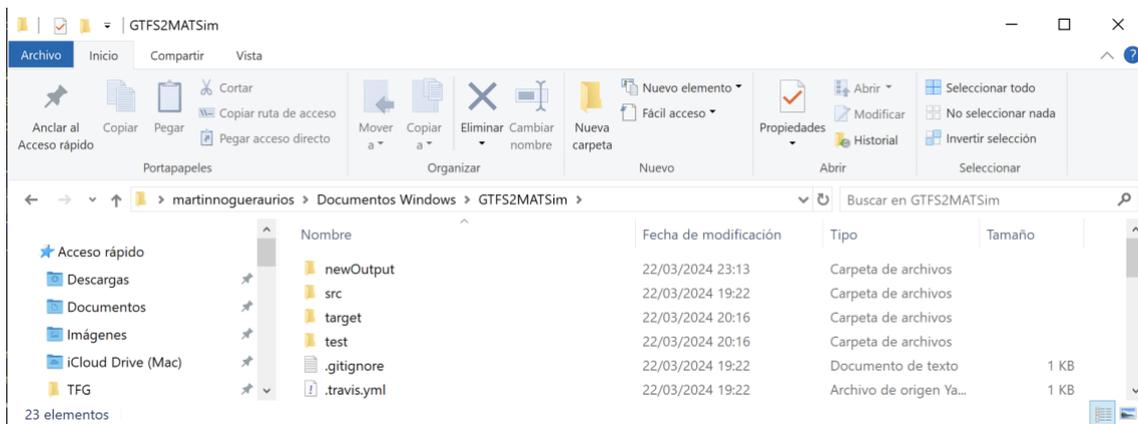


Ilustración 33. Clonación repositorio de GitHub GTFS2MATSim(3/3)

Una vez se tiene el proyecto GTFS2MATSim en la ubicación deseada, se va a llevar a cabo la manipulación y modificación de distintos archivos con VSCode para poder llevar a cabo la implementación de las paradas de autobuses en el proyecto. Lo cual se verá con detalle a continuación en el siguiente apartado. Todo esto formará parte del tratamiento y conversión de los datos.

4.3 DESARROLLO DE ALGORITMOS Y HERRAMIENTAS DE CONVERSIÓN

En esta sección, se detalla el proceso de desarrollo de algoritmos y herramientas necesarios para adaptar las bases de datos al formato requerido por MATSim. Las

siguientes subsecciones describen las scripts XML creadas y modificadas en Visual Studio Code para este propósito:

- **4.3.1 Creación de Scripts XML para la Conversión de Datos**

Se explican en detalle las scripts desarrolladas para la conversión de los datos de transporte público, incluyendo su estructura, funciones y su integración en la simulación.

Como se nombraba anteriormente, una vez clonado el repositorio GTFS2MATSim, se lleva a cabo la implementación de las paradas de autobuses mediante VSCode.

En primer lugar, se debe tener instalada la extensión de Maven for java en VSCode, para poder compilar el proyecto GTFS2MATSim.

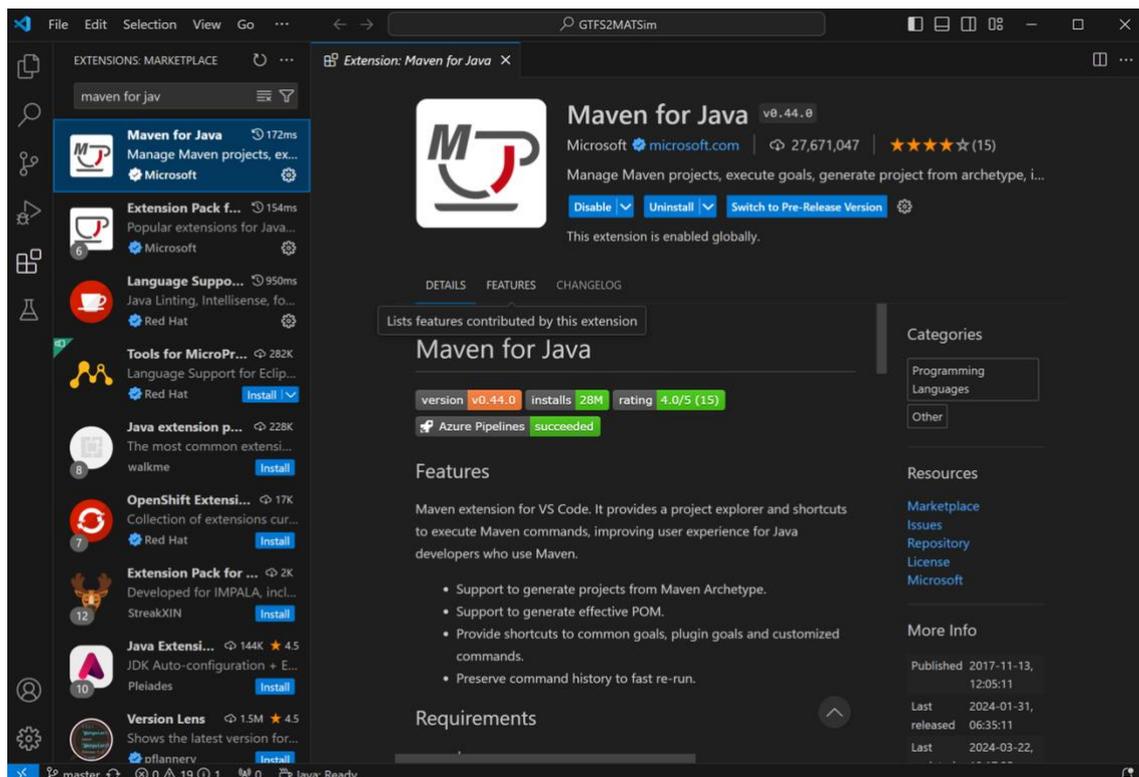


Ilustración 34. Extensión Maven en VSCode

El proyecto GTFS2MATSim como tal, posee archivos .java :

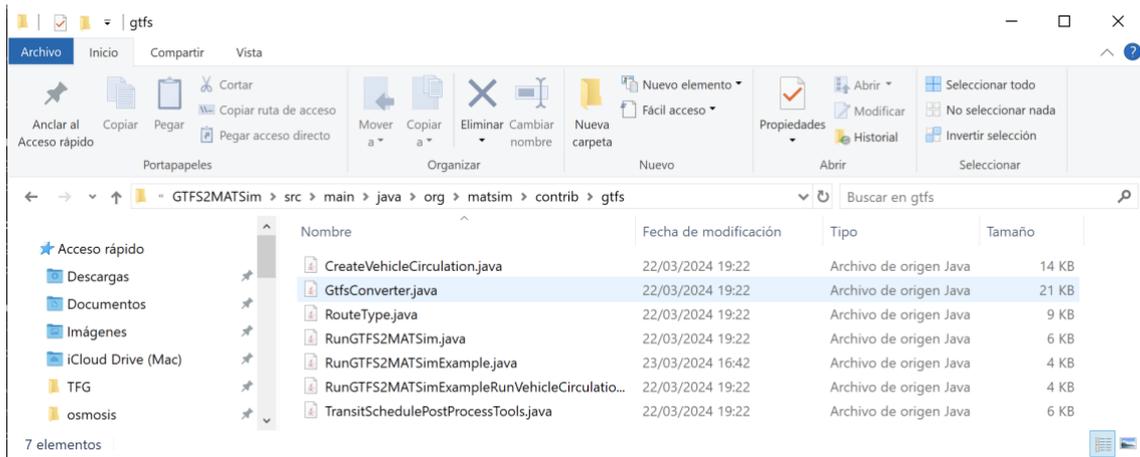


Ilustración 35. Archivos .java de Gtfs2Matsim

Pero no posee los archivos .class (clases compiladas de java) que necesitamos para llevar a cabo la correcta conversión a MATSim. Por lo que se necesita compilar el proyecto mediante el menú inferior de Maven en VSCode:

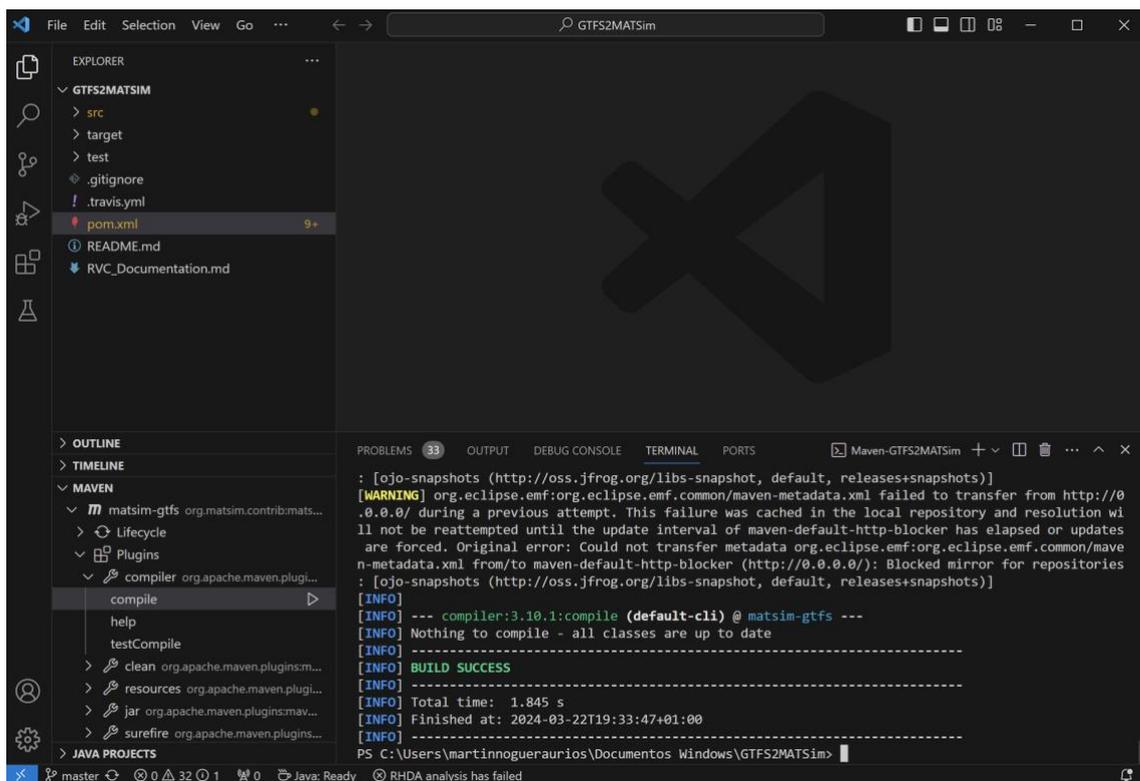


Ilustración 36. Compilación con Maven

Una vez compilado, se obtienen los archivos .class (clases compiladas de java) que se necesitan para llevar a cabo la correcta conversión a MATSim:

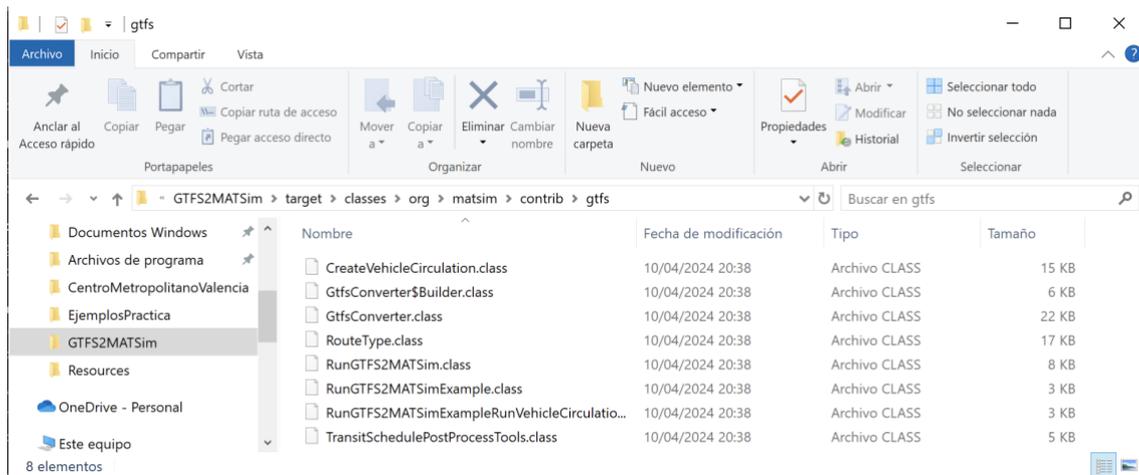


Ilustración 37. Archivos .class generados

Ahora ya se puede llevar a cabo la conversión de los archivos para adaptarlos a MATSim mediante el archivo “RunGTF52MATSimExample.java”, el cual está formado por una clase con un método el cual consiste en un algoritmo con diferentes partes:

- INPUT DATA:

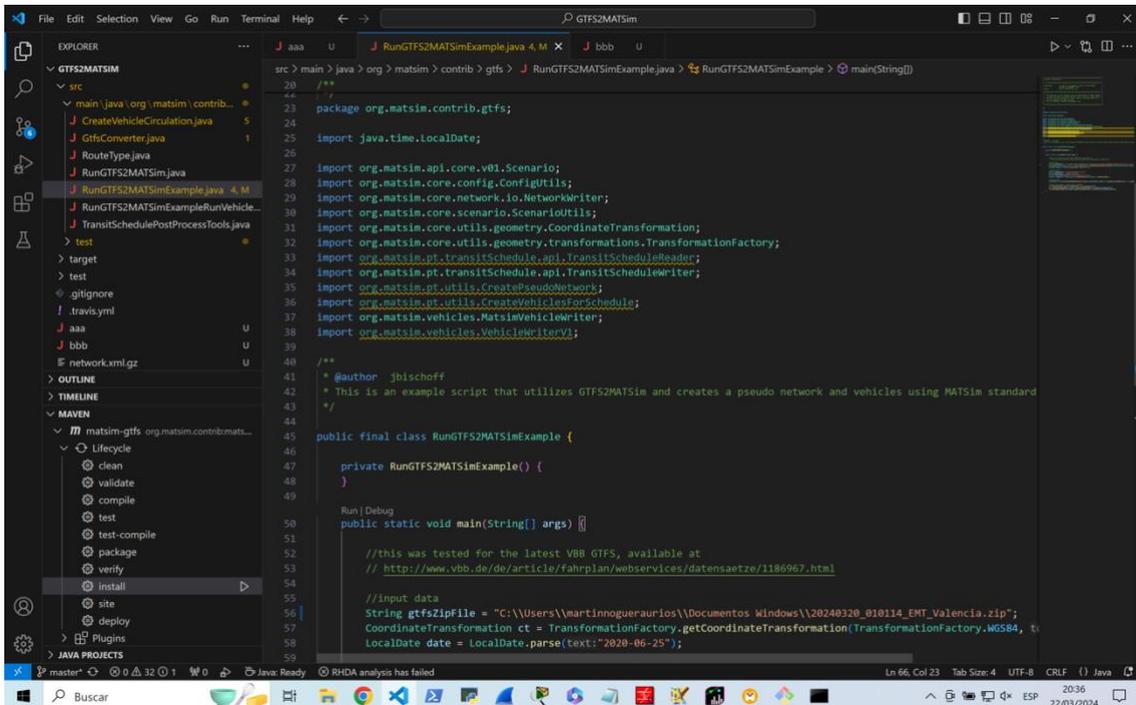
- Se indica la ubicación del archivo a convertir (el cual es el archivo.zip descargado de la web del Gobierno de España) [15] [16].
- Se elige una fecha (25-06-2020 en este caso, se quiere un día normal y corriente para los horarios de los buses)
- Coordenadas: realizará un cambio de coordenadas de WGS84 a EPSG:25830 (este parámetro no se modifica)

- OUTPUT DATA:

- La script devolverá tres archivos: una network.xml.gz, transitSchedule.xml.gz y transitVehicles.xml.gz. La network se utilizará puesto que se usará la creada por nosotros mismos.

Este proceso se hará tanto con el archivo .Zip de los autobuses urbanos, como con el .zip de los autobuses interurbanos, por lo que se obtendrá un Transit Schedule referido al urbano y otro al interurbano, y lo mismo con Transit Vehicles.

Aquí se puede observar la estructura de “RunGTFS2MATSimExample.java”

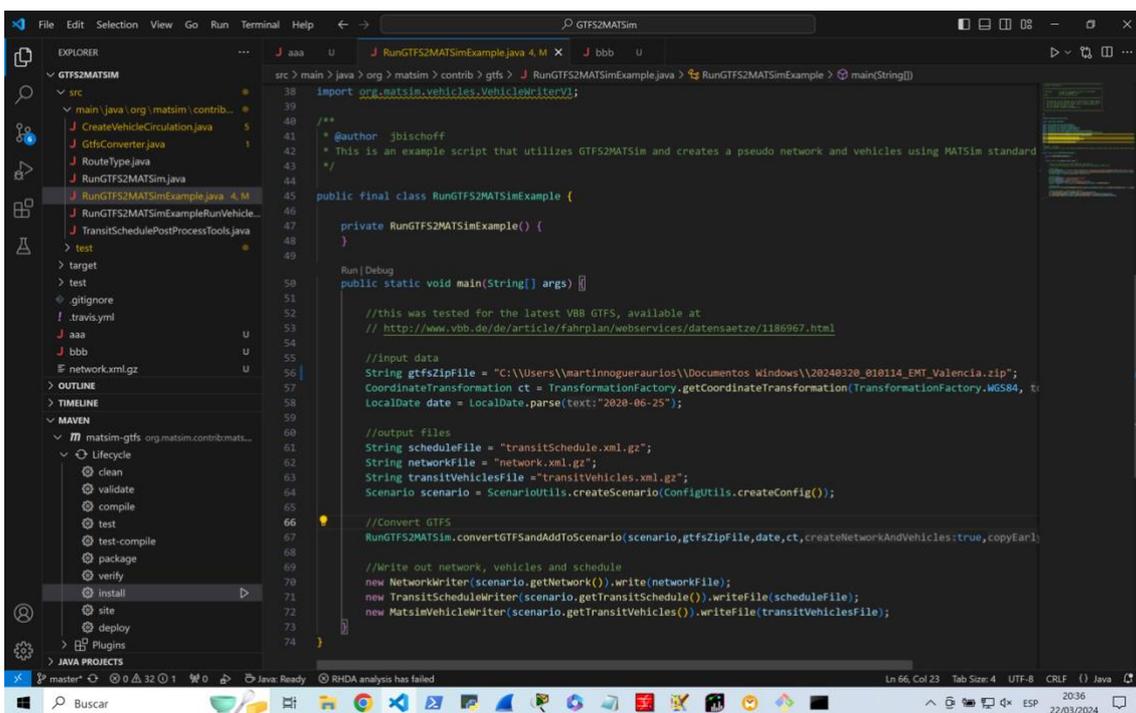


```

src > main > java > org > matsim > contrib > gtfs > J RunGTFS2MATSimExample.java > RunGTFS2MATSimExample > main(String[])
20 /**
21  *
22  */
23 package org.matsim.contrib.gtfs;
24
25 import java.time.LocalDate;
26
27 import org.matsim.api.core.v01.Scenario;
28 import org.matsim.core.config.ConfigUtils;
29 import org.matsim.core.network.io.NetworkWriter;
30 import org.matsim.core.scenario.ScenarioUtils;
31 import org.matsim.core.util.geometry.CoordinateTransformation;
32 import org.matsim.core.util.geometry.transformations.TransformationFactory;
33 import org.matsim.pt.transitSchedule.api.TransitScheduleReader;
34 import org.matsim.pt.transitSchedule.api.TransitScheduleWriter;
35 import org.matsim.pt.util.CreatePseudoNetwork;
36 import org.matsim.pt.util.CreateVehiclesForSchedule;
37 import org.matsim.vehicles.MatsimVehicleWriter;
38 import org.matsim.vehicles.VehicleWriterV1;
39
40 /**
41  * @author jbschoff
42  * This is an example script that utilizes GTFS2MATSim and creates a pseudo network and vehicles using MATSim standard
43  */
44
45 public final class RunGTFS2MATSimExample {
46
47     private RunGTFS2MATSimExample() {
48     }
49
50     Run [Debug]
51     public static void main(String[] args) {
52
53         //this was tested for the latest V88 GTFS, available at
54         // http://www.vbb.de/de/article/fahrplan/webservices/datensaetze/1186967.html
55
56         //input data
57         String gtfsZipFile = "C:\\Users\\martinnoguerauros\\Documents\\Windows\\20240320_010114_EMT_Valencia.zip";
58         CoordinateTransformation ct = TransformationFactory.getCoordinateTransformation(TransformationFactory.WGS84, t
59         LocalDate date = LocalDate.parse(text:"2020-06-25");
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Ilustración 38. Script RunGTFS2MATSimExample.java (1/2)



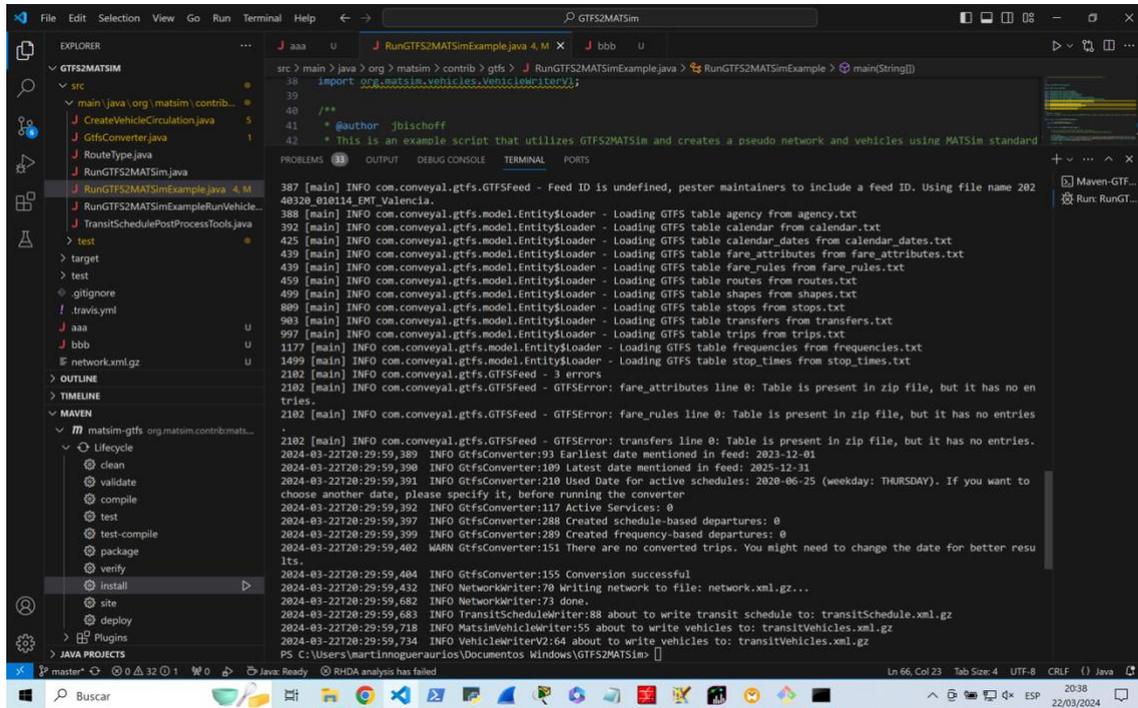
```

38 import org.matsim.vehicles.VehicleWriterV1;
39
40 /**
41  * @author jbschoff
42  * This is an example script that utilizes GTFS2MATSim and creates a pseudo network and vehicles using MATSim standard
43  */
44
45 public final class RunGTFS2MATSimExample {
46
47     private RunGTFS2MATSimExample() {
48     }
49
50     Run [Debug]
51     public static void main(String[] args) {
52
53         //this was tested for the latest V88 GTFS, available at
54         // http://www.vbb.de/de/article/fahrplan/webservices/datensaetze/1186967.html
55
56         //input data
57         String gtfsZipFile = "C:\\Users\\martinnoguerauros\\Documents\\Windows\\20240320_010114_EMT_Valencia.zip";
58         CoordinateTransformation ct = TransformationFactory.getCoordinateTransformation(TransformationFactory.WGS84, t
59         LocalDate date = LocalDate.parse(text:"2020-06-25");
60
61         //output files
62         String scheduleFile = "transitSchedule.xml.gz";
63         String networkFile = "network.xml.gz";
64         String transitVehiclesFile = "transitVehicles.xml.gz";
65         Scenario scenario = ScenarioUtils.createScenario(ConfigUtils.createConfig());
66
67         //Convert GTFS
68         RunGTFS2MATSim.convertGTFSandAddToScenario(scenario, gtfsZipFile, date, ct, createNetworkAndVehicles:true, copyEarl
69
70         //Write out network, vehicles and schedule
71         new NetworkWriter(scenario.getNetwork()).write(networkFile);
72         new TransitScheduleWriter(scenario.getTransitSchedule()).writeFile(scheduleFile);
73         new MatsimVehicleWriter(scenario.getTransitVehicles()).writeFile(transitVehiclesFile);
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Ilustración 39. Script RunGTFS2MATSimExample.java (2/2)

Una vez introducidos los INPUT DATA deseados, hay que ejecutar la script clickando en el menú de arriba a la derecha (el botón del triángulo):



```

src > main > java > org > matsim > contrib > gtfs > J RunGTFS2MATSimExample.java > RunGTFS2MATSimExample > main(String[])
16  import org.matsim.vehicles.VehicleLoaderV2;
40  /**
41   * @author jbischoff
42   * This is an example script that utilizes GTFS2MATSim and creates a pseudo network and vehicles using MATSim standard
387 [main] INFO com.conveyal.gtfs.GTFSFeed - Feed ID is undefined, pester maintainers to include a feed ID. Using file name 202
48320_010114_EMV_Valencia.
388 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table agency from agency.txt
392 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table calendar from calendar.txt
425 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table calendar_dates from calendar_dates.txt
439 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table fare_attributes from fare_attributes.txt
439 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table fare_rules from fare_rules.txt
459 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table routes from routes.txt
499 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table shapes from shapes.txt
889 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table stops from stops.txt
983 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table transfers from transfers.txt
997 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table trips from trips.txt
1177 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table frequencies from frequencies.txt
1499 [main] INFO com.conveyal.gtfs.model.EntityLoader - Loading GTFS table stop_times from stop_times.txt
2102 [main] INFO com.conveyal.gtfs.GTFSFeed - 3 errors
2102 [main] INFO com.conveyal.gtfs.GTFSFeed - GTFSError: fare_attributes line 0: Table is present in zip file, but it has no en
tries.
2102 [main] INFO com.conveyal.gtfs.GTFSFeed - GTFSError: fare_rules line 0: Table is present in zip file, but it has no entries
.
2102 [main] INFO com.conveyal.gtfs.GTFSFeed - GTFSError: transfers line 0: Table is present in zip file, but it has no entries.
2024-03-22T20:29:59,389 INFO GtfsConverter:93 Earliest date mentioned in feed: 2023-12-01
2024-03-22T20:29:59,390 INFO GtfsConverter:109 Latest date mentioned in feed: 2025-12-31
2024-03-22T20:29:59,391 INFO GtfsConverter:210 Used Date for active schedules: 2020-06-25 (weekday: THURSDAY). If you want to
choose another date, please specify it, before running the converter
2024-03-22T20:29:59,392 INFO GtfsConverter:117 Active Services: 0
2024-03-22T20:29:59,397 INFO GtfsConverter:288 Created schedule-based departures: 0
2024-03-22T20:29:59,399 INFO GtfsConverter:289 Created frequency-based departures: 0
2024-03-22T20:29:59,402 WARN GtfsConverter:151 There are no converted trips. You might need to change the date for better resu
lts.
2024-03-22T20:29:59,404 INFO GtfsConverter:155 Conversion successful
2024-03-22T20:29:59,432 INFO NetworkWriter:70 Writing network to file: network.xml.gz...
2024-03-22T20:29:59,682 INFO NetworkWriter:73 done.
2024-03-22T20:29:59,683 INFO TransitScheduleWriter:88 about to write transit schedule to: transitSchedule.xml.gz
2024-03-22T20:29:59,718 INFO MatsimVehicleWriter:55 about to write vehicles to: transitVehicles.xml.gz
2024-03-22T20:29:59,734 INFO VehicleWriterV2:64 about to write vehicles to: transitVehicles.xml.gz
PS C:\Users\martinnoqueauros\Documents\Windows\GTFS2MATSim>

```

Ilustración 40. Ejecución RunGTFS2MATSimExample.java

Una vez ejecutada, se encontrarán los 3 archivos de salida dentro de la carpeta GTFS2MATSim:

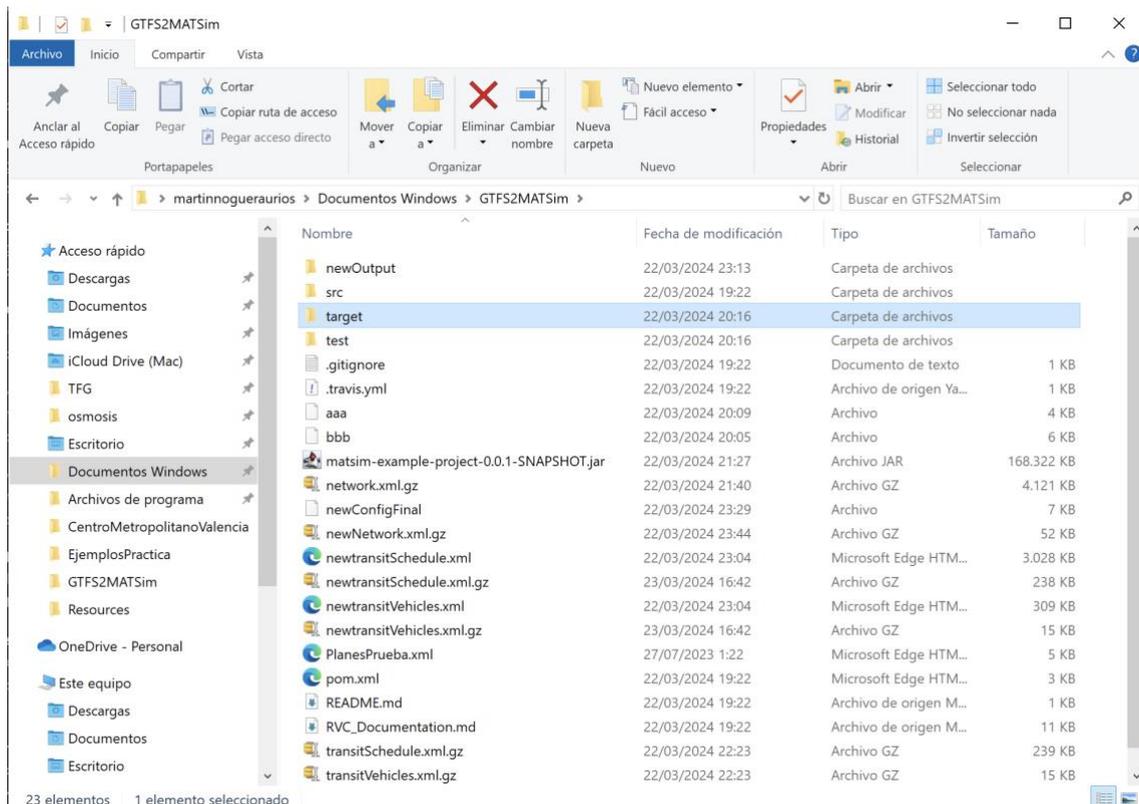


Ilustración 41. Archivos generados

En la captura aparecen varios debido a numerosas pruebas. Se observará la función de estos archivos generados en el apartado de Simulación.

A continuación, se va a llevar a cabo la explicación del contenido de una script Transit Schedule y de una script Transit Vehicles:

Transit Schedule:

Comienza <transitStops> donde define todas las paradas de autobús existentes, mediante un “id”, coordenadas cartesianas de su ubicación, el nombre de cada parada y el id del link de la network donde se sitúa cada una.

Dentro, tiene varias partes:

- <TransitLine id = “10---10”>: en este caso, está definiendo la línea 10.
- <transitRoute id="10---10_0">: se refiere al sentido de “ida” e indica que el modo de transporte es bus mediante <transportMode>bus</transportMode>
- <routeProfile> : Contiene una lista de paradas a lo largo de la ruta del autobús, formada cada una por:
 - **refId**: Identificador de la parada de autobús.

- **arrivalOffset** y **departureOffset**: Tiempos relativos de llegada y salida desde el inicio de la ruta.
- **awaitDeparture**: Un booleano que indica si el vehículo debe esperar hasta la hora programada de salida antes de continuar.
 - **<route>**: Lista de links de la network por los que la ruta pasa.
 - **<departures>**: formada por:
 - **id**: Identificador único para cada salida de la ruta.
 - **departureTime**: Hora de salida del vehículo.
 - **vehicleRefId**: Vehículo específico para esa salida.

Transit Vehicles:

Esta script comienza con lo siguiente:

```
<vehicleType id="defaultTransitVehicleType">
  <capacity seats="101" standingRoomInPersons="0">
  </capacity>
  <length meter="7.5"/>
  <width meter="1.0"/>
  <costInformation>
  </costInformation>
  <passengerCarEquivalents pce="1.0"/>
  <networkMode networkMode="car"/>
  <flowEfficiencyFactor factor="1.0"/>
</vehicleType>
```

Es decir, comienza creando el vehículo con todas sus características:

- Tiene una capacidad de 101 asientos, y nadie puede ir de pie.
- Mide 7.5m de largo y 1m de ancho.
- Puede operar en la red exactamente igual que un coche.

Y a continuación, contiene un extenso listado con la siguiente estructura cada una de las líneas: `<vehicle id="tr_0" type="defaultTransitVehicleType"/>`:

Cada elemento “vehicle” define un autobús con las características mencionadas anteriormente. Cada uno con un identificador “id” único, ya que podemos referenciar cada uno de manera diferente gracias a “defaultTransitVehicleType”.

Básicamente, esta script trata de crear todos los autobuses que se necesiten.

En conclusión, la adaptación y utilización del repositorio GTFS2MATSim para convertir datos GTFS a formato MATSim ha sido fundamental para el proyecto de modelización de movilidad urbana en Valencia. Al modificar un script Java en Visual Studio Code, se ha logrado una integración eficaz y precisa de los datos de transporte público en MATSim, optimizando el flujo de trabajo de preparación de datos y mejorando la precisión y funcionalidad del modelo de simulación. Este enfoque no solo facilita análisis más detallados y relevantes, sino que también establece una base sólida para futuras expansiones y aplicaciones del estudio de la movilidad urbana.

- **4.3.2 Modificación de Scripts Existente**

Se detalla cualquier modificación realizada en scripts XML existentes para adaptarlas a las necesidades específicas del proyecto, destacando los cambios realizados y su impacto en la preparación de los datos para la simulación.

En este apartado se va a tratar de explicar las scripts: config, plans, network y vehículos. Como ya se sabe, al clonar el repositorio matsim-example-project, teníamos un config.xml, plans100.xml y network.xml predeterminados, tal y como se muestra en la ilustración 28.

Los cuales han servido como una estructura para realizar los config, plans y network del proyecto:

NETWORK (CMV_2062.xml):

Una network describe la red de transporte sobre la cual los agentes van a viajar. Está compuesta por nodos (intersecciones) y enlaces o links (segmentos de carretera),

especificando atributos como la longitud, capacidad, número de carriles y límites de velocidad.

Sobre la network utilizada no hay nada más que añadir, puesto que no ha sido programada línea a línea, sino que se ha seguido todo un proceso explicado anteriormente hasta llegar a tener el archivo en formato xml y con la misma estructura que se acaba de mencionar.

CONFIG (configFinal.xml):

Un archivo config actúa como el archivo de configuración central para una simulación MATSim. Es esencial para iniciar y controlar cómo se ejecuta la simulación, qué datos se utilizan, y cómo se procesan los resultados. Es decir, sin config no hay nada.

Se procede a explicar los diferentes módulos que forman el configFinal:

- **Global:** valores predeterminados, con un sistema de coordenadas y permitiendo que se generen números aleatorios.
- **Network:** especifica el archivo de red que se utilizará en la simulación, en este caso CMV_2062.xml
- **Plans:** Define el archivo de entrada para los planes de los agentes (en este caso será planesPrueba.xml), que incluye las actividades y rutas de los agentes modelados, lo cual se verá con profundidad en el siguiente apartado.
- **Controler:** Aquí se especifica el directorio de salida cuando lo ejecute el Snapshot.jar, en este caso es la carpeta “output” del proyecto. El resto son valores por defecto.
- **Qsim:** tiene varios parámetros a mencionar:
 - **Vehicles Source:** fromVehiclesData significa que se utilizarán los datos de vehículos especificados en el archivo vehículos.xml. Y defaultVehicle significa que si no se especifica el vehículo, será un vehículo predeterminado para todos los modos.

· **Main Mode:** car, bicycle significa que la network soportará este tipo de vehículos.

· **Traffic Dynamics:** cómo modelar el tráfico. Kinematic Waves significa que se modela de forma más precisa la interacción entre vehículos.

- **PlanCalcScore:** La mayoría de valores se han dejado por defecto ya que no se va a llevar a cabo una puntuación para priorizar unas actividades frente a otras. Se han creado tres parámetros “modeParams” donde se define la circulación tanto en transporte público (pt), como en coche o bicicleta.

Cuando `marginalUtilityOfTraveling_util_hr` tiene un valor negativo, significa que existe un desgaste por pasar tiempo viajando en ese vehículo.

A continuación, se han definido diferentes parámetros con el valor “activityParams”, cada uno representa un tipo de actividad: home, work, work 24h, study, leisure, shop, shopping mall, hospital, univeristy. Todas las actividades están definidas con la misma prioridad, cada una tiene una duración típica y una mínima, así como una hora de inicio y una de fin.

- **PlansCalcroute:** Configura cómo se calculan las rutas para los modos de red especificados (coches y bicicletas). No hay ningún valor ya que se van a calcular todas las rutas de la misma manera.
- **Vehicles:** Define el archivo que contiene las características de los vehículos que se utilizarán en la simulación (en este caso, `vehiculos.xml`).
- **Strategy:** Define las estrategias que utilizarán los agentes para mejorar sus planes. No se ha modificado nada, se han dejado los valores por defecto.
- **TravelTimeCalculator:** Este módulo configura cómo se calculan los tiempos de viaje en la simulación de MATSim

`<param name="analyzedModes" value="car,bicycle" />` Especifica los modos de transporte para los cuales se deben calcular los tiempos de viaje. Es decir, se analizan tanto los coches como las bicicletas

`<param name="separateModes" value="true" />` Esto significa que la simulación mantendrá registros separados de tiempos de viaje para estos dos modos.

```
<?xml version="1.0" ?>
<!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
<config>

  <module name="global">
    <param name="randomSeed" value="4711" />
    <param name="coordinateSystem" value="Atlantis" /> <!--
"Atlantis"EPSG:25833-->
  </module>

  <module name="network">
    <param name="inputNetworkFile" value="CMV_2062.xml" />
  </module>

  <module name="plans">
    <param name="inputPlansFile" value="planesPrueba.xml" />
  </module>

  <module name="controller">
    <param name="outputDirectory" value="./output" />
    <param name="firstIteration" value="0" />
    <param name="lastIteration" value="10" />
  </module>

  <module name="qsim">

    <!-- "start/endTime" of MobSim (00:00:00 == take earliest
activity time/ run as long as active vehicles exist) -->
    <param name="vehiclesSource" value="fromVehiclesData" />
    <param name="mainMode" value="car,bicycle" />
    <param name="trafficDynamics" value="kinematicWaves" />
    <param name="startTime" value="00:00:00" />
    <param name="endTime" value="00:00:00" />
    <param name="vehiclesSource" value="defaultVehicle" />

    <param name="snapshotperiod" value="00:00:00"/> <!--
00:00:00 means NO snapshot writing -->
  </module>
</config>
```

```
</module>

<module name="planCalcScore">
  <param name="learningRate" value="1.0" />
  <param name="BrainExpBeta" value="2.0" />

  <param name="lateArrival" value="-18" />
  <param name="earlyDeparture" value="-0" />
  <param name="performing" value="+6" />
  <param name="waiting" value="-0" />

  <parameterset type="modeParams">
    <param name="mode" value="pt"/>
  </parameterset>

  <parameterset type="modeParams" >
    <param name="constant" value="0.0" />
    <param name="marginalUtilityOfDistance_util_m" value="0.0" />
    <param name="marginalUtilityOfTraveling_util_hr" value="-6.0"
/>
    <param name="mode" value="bicycle" />
    <param name="monetaryDistanceRate" value="0.0" />
  </parameterset>

  <parameterset type="modeParams">
    <param name="mode" value="car"/>
    <param name="marginalUtilityOfTraveling_util_hr" value="-
6.0" />
  </parameterset>

  <parameterset type="activityParams" >
    <param name="activityType"          value="h" /> <!-- home
-->
    <param name="priority"              value="1" />
    <param name="typicalDuration"       value="12:00:00" />
  </parameterset>
  <parameterset type="activityParams" >
    <param name="activityType"          value="w" /> <!-- work
-->
    <param name="priority"              value="1" />
    <param name="typicalDuration"       value="08:00:00" />
    <param name="openingTime"          value="07:00:00" />
    <param name="latestStartTime"      value="09:00:00" />
    <param name="earliestEndTime"     value="" />
    <param name="closingTime"          value="18:00:00" />
  </parameterset>

  <parameterset type="activityParams" >
```

```
        <param name="activityType"          value="w24" /> <!--
work 24h -->
        <param name="priority"              value="1" />
        <param name="typicalDuration"      value="08:00:00" />
    </parameterset>

    <parameterset type="activityParams">
        <param name="activityType"          value="study" /> <!--
study -->
        <param name="priority"              value="1" />
        <param name="typicalDuration"      value="05:00:00" />
        <param name="minimalDuration"     value="01:00:00" />
    </parameterset>

    <parameterset type="activityParams">
        <param name="activityType"          value="leisure" /> <!--
leisure -->
        <param name="priority"              value="1" />
        <param name="typicalDuration"      value="02:00:00" />
        <param name="minimalDuration"     value="00:20:00" />
    </parameterset>

    <parameterset type="activityParams">
        <param name="activityType"          value="shop" /> <!--
shop -->
        <param name="priority"              value="1" />
        <param name="typicalDuration"      value="00:40:00" />
        <param name="minimalDuration"     value="00:10:00" />
        <param name="openingTime"         value="08:00:00" />
        <param name="latestStartTime"     value="" />
        <param name="earliestEndTime"     value="" />
        <param name="closingTime"         value="20:00:00" />
    </parameterset>

    <parameterset type="activityParams">
        <param name="activityType"          value="cc" /> <!--
shopping mall -->
        <param name="priority"              value="1" />
        <param name="typicalDuration"      value="03:00:00" />
        <param name="minimalDuration"     value="01:00:00" />
        <param name="openingTime"         value="10:00:00" />
        <param name="latestStartTime"     value="" />
        <param name="earliestEndTime"     value="" />
        <param name="closingTime"         value="00:00:00" />
    </parameterset>

    <parameterset type="activityParams">
        <param name="activityType"          value="hosp" /> <!--
hospital (paciente o visita) -->
```

```
<param name="priority" value="1" />
<param name="typicalDuration" value="00:03:00" />
<param name="minimalDuration" value="00:01:00" />
<param name="openingTime" value="" />
<param name="latestStartTime" value="" />
<param name="earliestEndTime" value="" />
<param name="closingTime" value="" />
</parameterset>

<parameterset type="activityParams">
  <param name="activityType" value="uni" /> <!--
university (estudiante)-->
  <param name="priority" value="1" />
  <param name="typicalDuration" value="08:00:00" />
  <param name="minimalDuration" value="02:00:00" />
  <param name="openingTime" value="08:00:00" />
  <param name="latestStartTime" value="" />
  <param name="earliestEndTime" value="" />
  <param name="closingTime" value="21:30:00" />
</parameterset>

</module>

<module name="planscalcroute" >
  <param name="networkModes" value="car,bicycle" />
</module>

<module name="vehicles" >
  <param name="vehiclesFile" value="vehiculos.xml" />
</module>

<module name="strategy">
  <param name="maxAgentPlanMemorySize" value="5" /> <!-- 0 means
unlimited -->

  <param name="ModuleProbability_1" value="0.9" />
  <param name="Module_1" value="BestScore" />

  <param name="ModuleProbability_2" value="0.1" />
  <param name="Module_2" value="ReRoute" />
</module>

<module name="travelTimeCalculator" >
  <param name="analyzedModes" value="car,bicycle" />
  <param name="separateModes" value="true" />
</module>

</config>
```

PLANS (planesPrueba.xml):

Contiene los planes de los agentes, donde cada plan detalla las actividades diarias de un agente, incluyendo ubicaciones, tiempos y modos de transporte entre actividades.

Se procede a explicar un plan (persona 5) para tenerlo todo claro:

Se encuentra en su casa (ya que está realizando la actividad “h”), de la cual se tienen las coordenadas cartesianas, hasta las 8:30, momento en el que se dirigirá en bicicleta al trabajo (actividad “w”, también aparecen las coordenadas del trabajo, el cual se puede averiguar que es el Centro Comercial Arena) donde estará hasta las 17:30. En ese momento se dirigirá en bicicleta de vuelta a su casa (actividad “h”).

```

• <?xml version="1.0" ?>
• <!DOCTYPE plans SYSTEM
  "http://www.matsim.org/files/dtd/plans_v4.dtd">
• <plans xml:lang="es">
•
• <person id="persona1"> <!-- CASA-UPVcarrasca-CCaqua-CASA-->
•   <plan selected="yes">
•     <act type="h" link="314809207_1" x="885443.125"
y="546750.6875" end_time="8:30" />
•     <leg mode="car"/>
•
•     <act type="study" link="191615497_1" x="887287"
y="547689.8125" end_time="14:00"/>
•     <leg mode="car"/>
•
•     <act type="cc" link="820299471_1" x="887240" y="545052.125"
end_time="19:00"/>
•     <leg mode="car"/>
•
•     <act type="h" link="314809207_1" x="885443.125"
y="546750.6875" />
•   </plan>
• </person>
•
• <person id="persona2"> <!-- CASA-EDEM-comida-CONSUELO-CASA-->
•   <plan selected="yes">

```

```
• <act type="h" link="52092222_0" x="885222" y="545697.8125"  
end_time="8:30" />  
• <leg mode="car" />  
•  
• <act type="study" link="582629828_0" x="888819.125"  
y="545813.5625" end_time="13:00" />  
• <leg mode="car" />  
•  
• <act type="leisure" link="23425306_6" x="887540.625"  
y="545971.875" end_time="14:30" />  
• <leg mode="car" />  
•  
• <act type="hosp" link="12359038_0" x="883699.0625"  
y="545469.8125" end_time="18:00" />  
• <leg mode="car" />  
•  
• <act type="h" link="52092222_0" x="885222"  
y="545697.8125" />  
• </plan>  
• </person>  
•  
• <person id="persona3"> <!-- CASA-EUROPEA-CASA-hospMALVARROSA-  
CCsaler-CASA-->  
• <plan selected="yes">  
• <act type="h" link="23557400_0" x="882786.8125"  
y="546761.9375" end_time="8:30" />  
• <leg mode="car" />  
•  
• <act type="study" link="413952021_2" x="885489.8125"  
y="547158.5625" end_time="15:00" />  
• <leg mode="car" />  
•  
• <act type="h" link="23557400_0" x="882786.8125"  
y="546761.9375" end_time="17:00" />  
• <leg mode="car" />  
•  
• <act type="hosp" link="109837003_2" x="884758.0625"  
y="543687.5" end_time="18:30" />  
• <leg mode="car" />  
•  
• <act type="cc" link="39734853_3" x="886623.375"  
y="544823.25" end_time="21:00" />  
• <leg mode="car" />  
•  
• <act type="h" link="23557400_0" x="882786.8125"  
y="546761.9375" />  
• </plan>  
• </person>
```

```
• <person id="persona4"> <!-- CASA-CCarena-CASA-->
•   <plan selected="yes">
•     <act type="h" link="191715081_1" x="884162.875"
•     y="548922.1875" end_time="8:30" />
•     <leg mode="bicycle"/>
•
•     <act type="cc" link="84639055_2" x="885698.125"
•     y="549206.5625" end_time="18:00"/>
•     <leg mode="bicycle"/>
•
•     <act type="h" link="191715081_1" x="884162.875"
•     y="548922.1875" />
•   </plan>
• </person>
•
• <person id="persona5"> <!-- CASA-CCarenatrabajador-SALUD-CASA-->
•   <plan selected="yes">
•     <act type="h" link="12714863_1" x="883058.0625"
•     y="545474.375" end_time="8:30" />
•     <leg mode="bicycle"/>
•
•     <act type="w" link="84639055_2" x="885698.125"
•     y="549206.5625" end_time="17:30"/>
•     <leg mode="bicycle"/>
•
•     <act type="hosp" link="13947589_0" x="886875.875"
•     y="546566" end_time="22:00"/>
•     <leg mode="bicycle"/>
•
•     <act type="h" link="12714863_1" x="883058.0625"
•     y="545474.375" />
•   </plan>
• </person>
•
• <person id="persona6"> <!-- CASA-UVdarclases-CASA-PESET-ocio-CASA-->
•   <plan selected="yes">
•     <act type="h" link="143297509_0" x="885559.3125"
•     y="545171.0625" end_time="8:30" />
•     <leg mode="bicycle"/>
•
•     <act type="w" link="94118398_0" x="887233.0625"
•     y="547677.4375" end_time="14:30"/>
•     <leg mode="bicycle"/>
•
•     <act type="h" link="143297509_0" x="885559.3125"
•     y="545171.0625" end_time="16:00"/>
•     <leg mode="bicycle"/>
```

```
• <act type="hosp" link="23539840_1" x="883114.5625"  
y="544728.3125" end_time="17:30"/>  
• <leg mode="bicycle"/>  
•  
• <act type="leisure" link="11974877_3" x="883330.125"  
y="5445248.625" end_time="21:00"/>  
• <leg mode="bicycle"/>  
•  
• <act type="h" link="143297509_0" x="885559.3125"  
y="545171.0625" />  
• </plan>  
• </person>  
•  
• </plans>  
•  
• <!-- HOSPITALES (amarillo)  
•  
• 9 octubre link="500804645_0" x="882375.75" y="547360.875"  
•  
• CONSUELO link="12359038_0" x="883699.0625" y="545469.8125"  
•  
• QUIRON link="61722238_1" x="885579.625" y="547490.5625"  
•  
• CLINICO link="770904177_0" x="885842.0625" y="547471.4375"  
•  
• LA NUEVA FE link="109837003_2" x="884758.0625" y="543687.5"  
•  
• HOSPITAL GENERAL link="11936273_3" x="882007.875" y="546268.25"  
•  
• PESET link="23539840_1" x="883114.5625" y="544728.3125"  
•  
• IMED link="144013788_1" x="881733.625" y="550014.25"  
•  
• MALVARROSA link="13963491_4" x="888974.25" y="547256.375"  
•  
• CASA DE LA SALUD link="13947589_0" x="886875.875" y="546566"  
•  
• CENTROS COMERCIALES (ROSA)  
•  
• CC ARENA link="84639055_2" x="885698.125" y="549206.5625"  
•  
• CC EL SALER link="39734853_3" x="886623.375" y="544823.25"  
•  
• CC AQUA link="820299471_1" x="887240" y="545052.125"  
•  
• UNIVERSIDADES (AZUL)  
•  
• UPVprincipio link="191615498_2" x="887026.875" y="547775.5"  
• UPVcarrasca link="191615497_1" x="887250.6875" y="547701.75"
```

```
• UPVfinal link="84202763_2" x="887650.6875" y="547569.5625"  
•  
• UVprincipio link="94131883_1" x="887021.375" y="547747.875"  
• UVfin link="94118398_0" x="887233.0625" y="547677.4375"  
•  
• ESIC link="9272722_3" x="886700" y="547164.125"  
•  
• EUROPEA link="413952021_2" x="885489.8125" y="547158.5625"  
•  
• EDEM link="582629828_0" x="888819.125" y="545813.5625"  
•  
• -->
```

VEHICLES (vehículos.xml):

Define los tipos de vehículos disponibles en la simulación. Cada definición de vehículo puede incluir características como la capacidad de pasajeros, dimensiones físicas y otros parámetros relacionados con su desempeño y uso en la red.

En este caso hemos definido 2 tipos de vehículos: coche y bicicleta.

Coches: Tienen una longitud de 7.5m y un ancho de 2.0m, con una velocidad de 16.67 m/s

Bicicletas: Tienen una longitud de 2.0 m y un ancho de 1.0 m, con una velocidad de 4.17 m/s

Además, se adjudica a cada agente el tipo de vehículo que utilizará, aunque también se especifica al crear cada uno de los planes de cada agente.

```
• <?xml version="1.0" encoding="UTF-8"?>  
• <vehicleDefinitions xmlns="http://www.matsim.org/files/dtd"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.matsim.org/files/dtd  
  http://www.matsim.org/files/dtd/vehicleDefinitions_v2.0.xsd">  
•   <vehicleType id="car">  
•     <length meter="7.5"/>  
•     <width meter="2.0"/>  
•     <maximumVelocity meterPerSecond="16.67"/>  
•     <passengerCarEquivalent pce="1.0"/>
```

```
•
•
• </vehicleType>
•
•
• <vehicleType id="bicycle">
•   <length meter="2"/>
•   <width meter="1.0"/>
•   <maximumVelocity meterPerSecond="4.17"/>
•
•   <passengerCarEquivalents pce="0.25"/>
• </vehicleType>
•
•
•
• <vehicle id="persona1" type="car"/>
• <vehicle id="persona2" type="car"/>
• <vehicle id="persona3" type="car"/>
• <vehicle id="persona4" type="bicycle"/>
• <vehicle id="persona5" type="bicycle"/>
• <vehicle id="persona6" type="bicycle"/>
•
•
• </vehicleDefinitions>
•
```

Así, queda explicado cómo he abordado la conversión de datos y la preparación para la simulación utilizando las scripts XML, proporcionando detalles sobre su desarrollo y su papel en la metodología.

4.4 SOFTWARE COMPLEMENTARIO SIMUNTO VIA

Este apartado describe la integración y aplicación del software que complementa el modelado en MATSim, **SIMUNTO VIA**, un software utilizado para la visualización de datos y resultados de las simulaciones de transporte.

- **Creación de Simunto Via:**

Poco después de la fundación de Senozon AG, comenzaron las primeras presentaciones a clientes potenciales; la falta de material visual era un hándicap evidente. Explicar a los clientes que todas las respuestas a sus preguntas estaban contenidas en un enorme archivo

de eventos no era satisfactorio; las imágenes o incluso las animaciones les facilitaban mucho la comprensión. Así pues, nada más crearse la empresa se empezó a trabajar en una herramienta de visualización.

Aunque en un principio se planeó como una herramienta puramente interna, pronto quedó claro que una herramienta gráfica de visualización y análisis también beneficiaría a otros usuarios de MATSim. Tras una fase de pruebas beta con usuarios seleccionados de MATSim en la primavera de 2011, la primera versión de Via se lanzó en julio de 2011. Desde entonces, la aplicación ha crecido continuamente.

- **Descripción General:**

SIMUNTO VIA es un software avanzado de visualización y análisis que se utiliza principalmente para interpretar los resultados de simulaciones de movilidad y transporte realizadas con MATSim (Multi-Agent Transport Simulation).

MATSim es una plataforma de simulación de transporte basada en agentes que permite la modelización y análisis de políticas de transporte y movilidad a gran escala.

SIMUNTO VIA se integra no solo como una herramienta de visualización sino como parte integral de la metodología de análisis y validación.

- **Características:**

- **Visualización de la Red de Transporte:** SIMUNTO VIA permite cargar y visualizar detalladamente la red de transporte utilizada en MATSim (la network), mostrando rutas de transporte público, enlaces de red y nodos, así como las actividades de diferentes agentes (los planes), con sus rutas de viajes y el respectivo uso de diferentes modos de transporte.

A su vez, permite la visualización de patrones de movilidad, y otros elementos críticos en estudios de transporte. Los cuales pueden ser útiles para un posible futuro proyecto enfocado de una manera distinta.

- **Interfaz de Usuario:** Su interfaz es interactiva y está diseñada para facilitar la exploración y el análisis de grandes volúmenes de datos de simulación.

Posee distintos menús para los archivos, desde la función que van a desempeñar en la simulación hasta el diseño de la forma y color de cada uno de los agentes. Todo esto se verá con mayor detenimiento más adelante con una serie de grabaciones.

- **Análisis Temporal y Espacial:** Ofrece funcionalidades para observar cómo evolucionan las variables de tráfico a lo largo del tiempo y en diferentes partes de la red. En este caso, el transcurso de un día completo.

- **Análisis de Datos de Simulación:**

Proporciona herramientas para analizar los datos generados por las simulaciones de MATSim, facilitando la comprensión de los resultados a través de gráficos, mapas, y otros medios visuales.

- **Herramientas de Análisis de Tráfico:** Explica las herramientas específicas que ofrece SIMUNTO VIA para el análisis de flujos de tráfico, velocidades, y distancias. Ya que se trata de un entorno interactivo.

- **Integración con MATSim**

SIMUNTO VIA se integra perfectamente con MATSim, proporcionando un entorno en el que los usuarios pueden cargar directamente los archivos de salida de MATSim para su análisis visual.

- **Flujo de trabajo:** Todo comienza con la **preparación de los resultados**, ya que mediante el uso del ejecutable SNPASHOT.jar de MATSim, se lleva a cabo la ejecución de la simulación, ya que a partir de un archivo config.xml, el ejecutable devolverá una carpeta con una serie de archivos necesarios para la simulación y visualización en Simunto Via (output_events.xml entre ellos)

Entonces, en este instante se procede a la **carga de resultados** en Simunto Via, en el primer menú de todos (se verá todo en las grabaciones) para su posterior análisis y adjudicación de una función.

- **Visualización de Simulaciones:** Explica el proceso para iniciar la visualización de los resultados de simulación, incluyendo la configuración de parámetros iniciales y la selección de los datos de interés.

- **Ventajas de Usar SIMUNTO VIA en este Proyecto**

- **Interpretación de Datos:** SIMUNTO VIA facilita la interpretación de los complejos conjuntos de datos y resultados de simulación, ya que proporciona una visión muy simple de nuestros archivos. Esto sería imposible entender de forma tan eficaz únicamente leyendo uno a uno todas las líneas de código de los archivos.

- **Validación de Resultados:** Las visualizaciones de las simulaciones han contribuido en la validación de los modelos de tráfico desarrollados, permitiendo identificar errores de forma mucho más rápida que fijándose únicamente en el código programado [3].

Conclusión

SIMUNTO VIA es especialmente útil en la planificación urbana y la investigación del transporte, donde los resultados visuales pueden ayudar a comprender mejor los sistemas de movilidad y contribuir a una planificación más eficiente.

Todas estas herramientas y resultados visuales mencionados en la explicación del Software implementado serán demostrados mediante grabaciones y análisis en los siguientes apartados.

5. SIMULACIÓN DEL ESCENARIO Y RESULTADOS

- **Introducción:**

Se va a proceder a la simulación del escenario, realizado utilizando el software MATSim y complementado con visualizaciones a través de SIMUNTO VIA. El propósito fundamental de este apartado es demostrar cómo la conversión adecuada y eficaz de datos en formatos compatibles con herramientas de simulación puede proporcionar perspectivas valiosas sobre el comportamiento del sistema de transporte. Este proceso permite una visualización integral y un análisis de los efectos de diversas configuraciones de datos sobre la red de transporte de Valencia. Mediante grabaciones detalladas y visualizaciones, se presentan los resultados de la simulación, subrayando la técnica de manipulación y conversión de datos como un componente crucial en la modelización de sistemas de transporte complejos.

- **Grabaciones:**

Se va a explicar la funcionalidad de los diferentes menús que tiene el programa, así como el funcionamiento del escenario:

Como se puede observar en la interfaz del programa en el siguiente video:

https://drive.google.com/file/d/1ni94PWkEJF62rrRJYve1Jzm1V5iMCJC2/view?usp=drive_link

Simunto Via tiene 4 menús, el primero de todos, llamado DATA SOURCES, sirve para la carga de datos, es decir, se introduce en este menú todos los archivos que se quieran utilizar.

Para comenzar, se carga la network (CMV_2062.xml) y el output_events.xml.gz generado por el snapshot.jar en la carpeta output, el cual contiene todos los planes de los agentes (lo programado en planesPrueba.xml)

A continuación, podemos observar el segundo menú: LAYERS. Este menú sirve para adjudicar a cada archivo, la función que va a desempeñar en la simulación.

En este caso, se adjudica la función de network a CMV_2062.xml, de esta manera, ya se puede visualizar en el programa.

Además, a output_events.xml.gz en el menú de los agentes, en “Vehicles – from events”, se le adjudica la función de “eventos”, y se especifica que esto funcione sobre la network.

Una vez cargados los vehículos, se puede observar como en la parte inferior se carga el reloj con las horas del día.

Se aprecia como durante el transcurso del día, cada agente va realizando las tareas especificadas en el archivo planesPrueba.xml explicado anteriormente.

Cabe recalcar que se pueden crear tantos agentes y actividades como uno desee, al ser solo con un fin demostrativo, no se han creado más.

En este segundo menú LAYERS, se puede cambiar el color de la network para que resulte más cómodo trabajar. Se puede observar en el siguiente video:

https://drive.google.com/file/d/1TffH2tLaNnXJdNRNw3vR3BRaSBfAvZsl/view?usp=drive_link

Ahora, se va a explicar en qué consiste el cuarto menú: OVERLAYS.

https://drive.google.com/file/d/1YgHd6MpPQaw9xbCEExl87o4FS3nx2gXH/view?usp=drive_link

Se trata de un menú más interactivo, el cual no modifica la simulación. Según la finalidad del proyecto, se pueden incorporar o eliminar diferentes herramientas como un reloj, una leyenda que indica los archivos cargados y los modos de transporte permitidos, así como escalas y una flecha apuntando el norte. En este caso, se va a hacer uso del reloj, ya que lo demás no influye en la finalidad del proyecto.

El menú LAYERS tiene varias funciones o facilidades:

- Tocar sobre un agente y que muestre el recorrido que va a hacer a lo largo del día. Y no solo eso, sino que además se despliega un menú en la parte derecha de la

pantalla con los horarios de todos los desplazamientos entre actividades de su día (con eso se puede saber no solo el tiempo que pasa desplazándose, sino también el tiempo que pasa realizando cada actividad), indicando exactamente qué agente es (por ejemplo, la persona 3); además de sus atributos: la velocidad a la que está circulando, el id del agente, el tipo de vehículo en el que se desplaza, así como un indicador de si está en ruta en ese preciso instante:

https://drive.google.com/file/d/1urlB-0x4dJhC_DcVUG-192zgcCjCoq6s/view?usp=drive_link

- Cambiar la forma de los agentes, para así poder poner el tipo de vehículo que se necesite si se hace un estudio con muchos agentes, ya que es una forma de poder diferenciarlos con facilidad. Se pueden utilizar diferentes criterios de filtrado, se ha filtrado por el “network mode”, es decir, el modo por el que se desplaza en la network, se ha especificado car y bicycle y se les ha atribuido una forma distinta a cada uno. Como se puede observar en la visualización, queda mucho más claro.

https://drive.google.com/file/d/1ZnhJo-OKtBUkWW3S0O7iKLUr6NzI1bTr/view?usp=drive_link

- Ver una gráfica con las estadísticas de cuantos vehículos en ruta hay en cada momento del día.

https://drive.google.com/file/d/1F_cb0Yp-TsPbPdTd-3V4se3o9ZSaro3d/view?usp=drive_link

- Clickar sobre nodo en concreto y que se indique su id, las coordenadas cartesianas en las que se sitúa y cuantos links pasan por él.
- Clickar sobre un link y que proporcione su id, sus coordenadas cartesianas, de que nodo a que nodo va dicho link, los modos de transporte permitidos (esto serviría para poder implementar un carril bici si se quisiese enfocar de esa manera el proyecto, por ejemplo).

También muestra otras características de cada link como el número de carriles que tiene, la velocidad permitida, su longitud y su capacidad.

https://drive.google.com/file/d/11oRDB6UZ5sGOU2d0IYesN6bL6n89YLLWW/view?usp=drive_link

TRANSPORTE PUBLICO:

Al igual que se ha cargado la network y los planes, se van a cargar los archivos relacionados con los autobuses en el menú DATA SOURCES, tanto el urbano como el interurbano, cada uno de ellos está formado por un TransitSchedule.xml y un TransitVehicles.xml, por lo que se cargarán cuatro archivos.

Una vez cargados, en el menú LAYERS, en el apartado de Transit, se adjudica a cada uno su función (Schedule o vehicles). De esta manera se puede observar todas las paradas de los autobuses, pudiendo distinguir entre el urbano y el interurbano.

https://drive.google.com/file/d/151COs7UbD2FzDOKtooboQG7leIlTHPo5/view?usp=drive_link

Continuando con funciones que proporciona el menú LAYERS para los autobuses:

- Ver el nombre de cada una de las paradas de autobuses de Valencia:
https://drive.google.com/file/d/1NC9OjLW0OEIH7T_XpH_MK7aFPTKX9iOy/view?usp=drive_link
- Al marcar en favorito alguna parada, se nos mostrará (y se quedarán fijadas) todas las líneas de autobús (con todas sus paradas) que pasan por dicha parada. Además, se puede cambiar el color de la línea y de las paradas al gusto del consumidor, lo cual puede servir para llevar a cabo algún tipo de estudio, ya que es una forma eficaz de distinguir a simple vista unas líneas de otras.
- Y no solo eso, sino que además se abre un menú desplegable a la derecha especificando todas líneas, una a una, con sus respectivas rutas (sentido).

https://drive.google.com/file/d/1p2qLAHl9Wb4NsIj48XYHHtdBLx1Oka7I/view?usp=drive_link

https://drive.google.com/file/d/1u8G6GN2fqNG9JAAtX4P5M0DICB-GUA9xu/view?usp=drive_link

- Por otro lado, si no se marca como favorito la parada seleccionada, no se quedarán fijadas todas las rutas que pasan por ella, por lo que también sirve para ir visualizando una a una las paradas y rutas necesarias

https://drive.google.com/file/d/10VTB5OTKnbYMbdV-6NmdjrinxfkE2x7M/view?usp=drive_link

Conclusión

A través de un escenario de simulación meticulosamente configurado, se ha demostrado que la correcta preparación de los datos puede desbloquear un potencial significativo para entender y mejorar la planificación del transporte. Las grabaciones y visualizaciones incorporadas ilustran no solo la aplicabilidad de los datos convertidos en contextos de simulación, sino también cómo estos pueden influir en la toma de decisiones basada en evidencia. Este enfoque detallado confirma la viabilidad del método empleado y proporciona una base sólida para futuras investigaciones que busquen optimizar la modelización de transporte urbano y la planificación de políticas a partir de datos complejos.

Este estudio, por lo tanto, no solo valida la técnica de conversión de datos utilizada, sino que también establece un precedente para exploraciones futuras en la modelización avanzada de la movilidad urbana.

6. DISCUSIÓN

Este estudio ha permitido una interpretación detallada de cómo la modelización basada en agentes, implementada a través de MATSim, puede efectivamente simular la movilidad urbana en Valencia.

- **Aplicabilidad de la metodología en otros contextos:**

Los hallazgos destacan la aplicabilidad de la metodología no solo en Valencia sino también en otros contextos urbanos o ciudades, demostrando su flexibilidad y adaptabilidad.

Las técnicas de modelización basadas en agentes pueden adaptarse para reflejar diferentes configuraciones urbanas y patrones de movilidad, lo que las hace extremadamente versátiles para estudios urbanos más amplios.

Esta investigación ha establecido un precedente metodológico importante para proyectos futuros en el ámbito de la movilidad urbana, ofreciendo un marco replicable que puede adaptarse a diferentes necesidades y entornos.

- **Posibles Mejoras para Futuras Investigaciones:**

Para futuros estudios, sería beneficioso integrar un enfoque más detallado en el análisis de los patrones de movilidad individual y explorar en mayor profundidad cómo las diferencias socioeconómicas afectan las decisiones de movilidad. Además, la incorporación de datos en tiempo real podría mejorar la precisión de las simulaciones y permitir una planificación más reactiva y dinámica.

7. CONCLUSIONES

En la conclusión de este proyecto sobre la modelización de la movilidad urbana en Valencia mediante el uso de MATSim, se enfatiza la integración exitosa de una metodología detallada para analizar la planificación del transporte. Este estudio no solo ha demostrado la viabilidad de aplicar la modelización basada en agentes en un contexto urbano complejo, sino también ha ilustrado cómo dichas herramientas pueden influir significativamente en la toma de decisiones estratégicas para el desarrollo urbano y la movilidad sostenible.

- **Conclusión del Estudio:**

Este proyecto ha consolidado una metodología robusta que combina la recopilación meticulosa de datos, su tratamiento y conversión adecuada para simulaciones, y el uso de MATSim para crear escenarios de simulación realistas.

Para llevar a cabo esta técnica de modelización, se ha desarrollado un entorno que incluye la configuración y uso de herramientas como Java, Maven, osmosis, JOSM y GitHub.

Además, se han creado algoritmos específicos para adaptar los datos a MATSim y utilizado un software complementario para la visualización de los resultados.

La metodología que se ha seguido también implica la recopilación, tratamiento y conversión de datos a formatos adecuados para su análisis y uso en el software de simulación Simunto Via.

Los resultados obtenidos han permitido una visualización de un funcionamiento real del transporte, así como sentar las bases para poder mejorar la eficiencia y sostenibilidad del sistema de transporte de Valencia.

- **Recomendaciones:**

Se recomienda la continuación del desarrollo de modelos de simulación más detallados que integren tecnologías emergentes como el Big Data y el aprendizaje automático, lo cual podría mejorar la precisión de las simulaciones y permitir una planificación más reactiva y dinámica. Además, fomentar la colaboración interdisciplinaria entre urbanistas, ingenieros de transporte, y científicos de datos para abordar los desafíos de movilidad de manera integral.

Este proyecto no solo contribuye al campo de la planificación del transporte y la movilidad urbana, sino que también sirve como un recurso valioso para las autoridades

locales y planificadores urbanos en su esfuerzo por crear ciudades más habitables, eficientes y sostenibles.

8. BIBLIOGRAFÍA

- [1] Generalitat Valenciana, Conselleria de Política Territorial, Obres Públiques i Mobilitat. “Plan de Movilidad del Área Metropolitana de Valencia”, (January 2022)
https://www.pmomevalencia.com/_files/ugd/e0f03f_8e66466799ba4db9a2ea31b92c7349e3.pdf
- [2] PTV GROUP. (n.d.). *Modelización del transporte* / PTV Group.
Www.ptvgroup.com. Consultado el 6 mayo de 2024, de
<https://www.ptvgroup.com/es/areas-de-aplicacion/modelizacion-del-transporte>
- [3] MATSim Association, Mats. C. (2024). *About MATSim*. MATSim.org. Consultado en marzo de 2024, de
<https://www.matsim.org/about-matsim>
- [4] Horni, A., Nagel, K., & Axhausen, K. (2024). The Multi-Agent Transport Simulation MATSim. In *MATSim (Multi-Agent Transport Simulation)*.
<https://www.matsim.org/files/book/partOne-latest.pdf>
- [5] Janek Laudan, entre otros contribuidores (2024). *matsim-org/matsim-example-project*. GitHub. Consultado en enero de 2024, de
<https://github.com/matsim-org/matsim-example-project>
- [6] IBM. (1 enero 2021). *IBM Docs*. [Www.ibm.com](http://www.ibm.com). Consultado en Marzo de 2024, de
<https://www.ibm.com/docs/es/i/7.3?topic=platform-java-development-kit>
- [7] Yagüe, C. (29 abril 2019). *Qué es Apache Maven*. OpenWebinars.net. Consultado en abril de 2024, de <https://openwebinars.net/blog/que-es-apache-maven/>

[8] ES:Osmosis - OpenStreetMap Wiki. (30 enero 2021). Wiki.openstreetmap.org.
<https://wiki.openstreetmap.org/wiki/ES:Osmosis#:~:text=Osmosis%20es%20una%20aplicaci%C3%B3n%20Java,llevar%20a%20cabo%20operaciones%20complejas>

[9] Scholz, I. (4 de abril 2024). *JOSM*. Wikipedia. <https://es.wikipedia.org/wiki/JOSM>

[10] ¿Cómo aplicar un correcto tratamiento de datos durante el proceso de transformación digital? (23 febrero 2022). Consultado en abril 2024, de <https://www.docuSign.com/es-mx/blog/tratamiento-de-datos#:~:text=Tratamiento%20de%20datos%3A%20%C2%BFQu%C3%A9%20es>

[11] Flores, N. (n.d.). *Recolección de datos: qué es, ventajas y consejos para usarlos*. Blog.maestriasydiplomados.tec.mx. Consultado el 19 de abril de 2024, de <https://blog.maestriasydiplomados.tec.mx/recoleccion-de-datos-que-es-ventajas-y-consejos-para-usarlos#:~:text=Entonces%20podemos%20decir%20que%20la>

[12] *Geofabrik Download Server*. (n.d.). Download.geofabrik.de. Consultado el 6 de Marzo de 2024, de <https://download.geofabrik.de/europe/spain/valencia.html>

[13] *Export*. (n.d.). OpenStreetMap. Retrieved March 6, 2024, from <https://www.openstreetmap.org/export#map=16/39.4793/-0.3627&layers=T>

[14] Henderson, B. (n.d.). *Releases · openstreetmap/osmosis*. GitHub. Consultado el 9 de marzo de 2024, de <https://github.com/openstreetmap/osmosis/releases>



[15] Mitma. (2021). *Ficheros*. Transportes.gob.es. (Transporte Interurbano)

<https://nap.transportes.gob.es/Files/Detail/1325>

[16] Mitma. (2023). *Ficheros*. Transportes.gob.es. (Transporte Urbano)

<https://nap.transportes.gob.es/Files/Detail/965>

[17] Nico Kühnel, entre otros contribuidores (2024, April 10). *matsim-*

org/GTFS2MATSim. GitHub. <https://github.com/matsim-org/GTFS2MATSim>