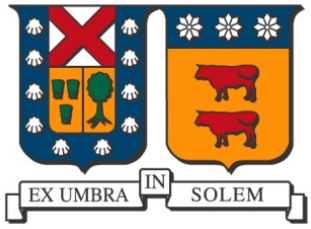
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“ALGORITMOS PARA DECISIONES DURANTE UNA EVACUACIÓN: CASO DE ESTUDIO VIÑA DEL MAR”

BASTIÁN ESTEBAN VIVAR MALDONADO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: María Cristina Riff Rojas
Profesor Correferente: Elizabeth Montero

Mayo - 2024

RESUMEN

Resumen—

En la siguiente memoria se estudia la red de transporte vehicular para las comunas de Viña del Mar y Valparaíso para el caso de la catástrofe natural de un tsunamis, para aquello se mencionan cuales fueron las consideraciones para modelar la red como un grafo dirigido en una primera instancia, utilizando las calles o arcos como unidireccional o bidireccional según las reglas de tránsito ya establecidos actualmente, generando también atributos de costo y capacidad para estos arcos utilizando información de dataset *open source* presente en la librería OSMnx. Luego se pobló la red utilizando datos demográficos del Censo y obteniendo la población de cada cuadra, al igual que asignar esta población a vehículos utilizando la información del parque vehicular de sus comunas respectivas otorgada por la INE. Una vez obtenido la distribución de vehículos en la red, se da paso a optimizar las rutas de aquellos vehículos con el fin de minimizar el tiempo de evacuación utilizando algoritmos iterativos, dando paso así a un análisis exploratorio de los resultados de aquella evacuación y pruebas de estres de la red original. Finalmente se entrega una propuesta de red especializada a nuestro objetivo, con el fin de generar una guía óptima para las entidades y así direccionar el tráfico fuera de las reglas establecidas minimizando el tiempo de evacuación de la población. Además de obtener información critica sobre la estructura tanto como sus puntos débiles y fortalezas.

Palabras Clave— Evacuation planning, Routing, Geographic Information Systems (GIS).

GLOSARIO

ABEM: *Agent Base Evacuation Model*
ABMs: *Agent Base Models*
CA: *Cellular Automata*
CEP: *Community Evacuation Problem*
ERP: *Evacuation Route Planning*
INE: Instituto Nacional de Estadística
SENAPRED: El Servicio Nacional de Prevención y Respuesta ante Desastres
STCM: *Suffix-Tree Conflict Minimazation*
STN: *Spatiotemporal Network*
ORCA: *Optimal Recipro-cal Collision Avoidance*
OSM: *Open Street Map*
OSMnx: *Open Street Map networkx*

ÍNDICE DE CONTENIDOS

RESUMEN i

GLOSARIO ii

ÍNDICE DE FIGURAS v

ÍNDICE DE TABLAS v

INTRODUCCIÓN **1**

1 Objetivos **3**

CAPÍTULO 2: DEFINICIÓN DEL PROBLEMA **3**

CAPÍTULO 3: Marco Conceptual **5**
- 3.1 *Modelos Heurísticos* . . . 7
- 3.2 *Network Flow Optimization methods* . . . 7
  - 3.2.1 *Conflict Free Evacuation Route Planning* (CF-ERP) . . . 7
  - 3.2.2 *Community Evacuation Problem* (CEP) . . . 8
  - 3.2.3 *Spatiotemporal Network* (STN) . . . 9
- 3.3 *Simulation Models* . . . 10
  - 3.3.1 ABMs . . . 11
  - 3.3.2 *Agent Based Evacuation Model*(ABEM) . . . 12

CAPÍTULO 4: PROPUESTA DE SOLUCIÓN **13**
- 4.1 Zona de peligro . . . 16
- 4.2 Estimación de población . . . 17
  - 4.2.1 Metodología para la Generación de Polígonos de Voronoi . . . 18
  - 4.2.2 Cálculo del Diagrama de Voronoi: . . . 19
  - 4.2.3 Manejo de Límites: . . . 19
- 4.3 Parque vehicular Valparaíso y Viña del Mar: . . . 23
  - 4.3.1 Asignación de Vehículos a nuestra red: . . . 25
- 4.4 Representación de solución: . . . 30
  - 4.4.1 Creación de Rutas . . . 30
- 4.5 Simulación Evacuación: . . . 32
  - 4.5.1 Organización de las rutas . . . 32
- 4.6 Realización de movimientos . . . 35
  - 4.6.1 Inicialización de Variables . . . 36
  - 4.6.2 Bucle Principal . . . 37
  - 4.6.3 Iteración por Marcas de Tiempo . . . 37
  - 4.6.4 Finalización de la Evacuación . . . 37
  - 4.6.5 Detalles de Evacuación . . . 37
- 4.7 Flujo Algoritmo Iterativo . . . 40

4.8 Análisis de resultados, mejora de rutas 42
4.8.1 Red de Transporte Original 43
4.9 Mejoras a la red de Transporte 49
4.9.1 Modificación de Red de transporte 53

CAPÍTULO 5: CONCLUSIONES **68**

REFERENCIAS BIBLIOGRÁFICAS **70**

Índice de figuras:

ÍNDICE DE FIGURAS

1 Red simple, ERP 6
2 Ejemplo CF-ERP [Herschelman y Yang, 2019] 8
3 STN, expansión temporal 9
4 Diagrama CEP 10
5 Red de transporte 14
6 Zona de Peligro 16
7 Polígonos de Voronoi 18
8 Cuadras escala logarítmica de población 21
9 Voronoi según Población, Zoom Placeres 22
10 Voronoi según Población (Escala logarítmica) 23
11 Costo de Evacuación Real 43
12 Costo de Evacuación Real Centro de Viña 44
13 Distribución Costo Real Vehículos 45
14 Costo de Evacuación Teórico, sin atascamientos 46
15 Distribución de Costo Teórico, sin atascamientos 47
16 Diferencia Proporcional, Costo Real / Costo Teórico 48
17 Zoom Diferencia Proporcional, Costo Real / Costo Teórico 49
18 Calles con Conflictos de Capacidad (Rojo) Viña del Mar 50
19 Calles con Conflictos de Capacidad (Rojo) Valparaíso 51
20 Utilización de Arcos 52
21 Cantidad de Vehículos por arco 53
22 Red Original 54
23 Arcos Modificados 57
24 Arcos Modificados Viña del Mar 58
25 Red Optimizada 59
26 Costo de Evacuación Real, Red Optimizada 60
27 Costo de Evacuación Real Zoom, Red Optimizada 60
28 Distribución Costo de Evacuación Real Zoom, Red Optimizada 61
29 Costo de Evacuación Teórico, Red Optimizada 62
30 Distribución Costo de Evacuación Teórico 62
31 Costo de Evacuación Proporcional 63
32 Costo de Evacuación Proporcional Zoom Viña del Mar 64
33 Distribución Costo de Evacuación Teórico Zoom 65
34 Calles de Salida Red Optimizada 66
35 Utilización de Calles Red Optimizada 66
36 Utilización de Calles Red Optimizada Zoom Viña del Mar 67

ÍNDICE DE TABLAS

1 Highway Capacity 15
2 Datos de vehículos para Viña del Mar y Valparaíso 24

3 Capacidad de asientos, longitud y valores adicionales de vehículos 25
4 Datos de Paraderos 27
5 Datos de stops_times 28

INTRODUCCIÓN

La vulnerabilidad de las ciudad costera de Viña del Mar ante la amenaza de tsunamis constituye un desafío crítico que requiere una planificación y respuesta meticulosas. Chile, ubicado en una de las zonas sísmicas más activas del mundo, enfrenta significativos riesgos de terremotos y tsunamis. En este contexto, la capacidad de evacuar rápidamente a los residentes es esencial para salvar vidas. Los tsunamis presentan desafíos únicos debido a su naturaleza impredecible y rápida, así como a las características geográficas y demográficas específicas de estas localidades. Esta memoria se enfoca en el desarrollo y análisis de algoritmos de evacuación que optimicen el uso de la red de transporte urbano de Viña del Mar , considerando la distribución demográfica y la simulación del comportamiento vehicular, así como la representación precisa de la red de transporte.

El objetivo principal de esta investigación es evaluar la efectividad de diferentes algoritmos de evacuación en estas ciudades costeras, mediante la simulación de escenarios de evacuación masiva. Para lograr esto, se modelará la red de transporte urbano, que incluye carreteras, intersecciones y otros elementos de infraestructura crítica, así como la distribución espacial de la población y además incorporando esta población en diferentes medios de transporte utilizando los parques vehiculares de la región. Utilizando herramientas de simulación y análisis de redes, se investigará cómo los flujos vehiculares pueden ser gestionados de manera más eficiente durante una emergencia por tsunami, esto con el fin de obtener zonas críticas y obtener los posibles comportamientos que puedan existir, generando así un análisis de estrés de la red.

El modelado de la red de transporte urbano incluirá un modelo detallado que incorporará datos geográficos y de infraestructura vial, sirviendo como base para la simulación de los flujos de tráfico durante una evacuación. La distribución demográfica se mapeará e integrará en el modelo de transporte para reflejar con precisión las áreas de alta densidad poblacional que requieren una atención especial durante la evacuación. La simulación del comportamiento vehicular permitirá modelar el movimiento de vehículos en la red de transporte durante un evento de evacuación, analizando los comportamientos de los arcos.

El análisis de la red de transporte se centrará en identificar los puntos fuertes y débiles, evaluando cómo los cuellos de botella y las áreas de alta congestión pueden afectar la efectividad de la evacuación. Se desarrollarán y probarán algoritmos de evacuación basados en diferentes estrategias, como rutas más cortas, desvíos predefinidos y control dinámico del tráfico. Estos algoritmos serán evaluados en términos de tiempo de evacuación, atascamiento y eficiencia.

Adicionalmente, se explorarán diversas modificaciones a la red de transporte existente con el fin de aumentar la eficiencia de la evacuación. Estas modificaciones incluirán la reconfiguración de rutas, la implementación de cambios de direcciones y la optimización de los puntos de salida y entrada en las principales arterias de la ciudad. Mediante la simulación de estos

cambios, se evaluará su impacto en el tiempo total de evacuación y la reducción de congestionamientos críticos, proporcionando un enfoque integral para mejorar la respuesta ante tsunamis.

Las recomendaciones serán bajo la hipótesis de que al buscar la mejor ruta posible se somete la red a un estrés dejando a la luz los puntos débiles y fuertes de la red de transporte, en conjunto con la generación de una nueva red específica, que utiliza las mismas calles ya establecidas variando así únicamente las direcciones establecidas dentro de la red.

Para abordar este complejo problema, inicialmente se desarrollarán los pasos necesarios para generar la red de transporte junto con todos los atributos requeridos para el análisis posterior. Una vez obtenida esta red, se procederá a poblarla asignando la población y los vehículos correspondientes, lo que permitirá generar un algoritmo capaz de producir rutas óptimas para el plan de evacuación. Posteriormente, se diseñará una nueva red especializada para emergencias, con el objetivo de proporcionar un plan de tráfico efectivo para las entidades responsables de la gestión de emergencias. Esta red optimizada incluirá modificaciones específicas para mejorar la eficiencia de evacuación, como la reconfiguración de rutas y la implementación de cambios de dirección a las calles, adaptadas a las necesidades críticas durante un evento de tsunami.

El resultado esperado de esta investigación es proporcionar una comprensión más profunda de cómo las redes de transporte urbano de Viña del Mar pueden ser optimizadas para mejorar las estrategias de evacuación en situaciones de emergencia. Además, se pretende ofrecer recomendaciones prácticas para los planificadores urbanos y responsables de la gestión de emergencias, con el fin de fortalecer la resiliencia y conocimientos de los puntos claves de estas dos comunas. Este estudio contribuirá a una mejor preparación y respuesta ante emergencias, generando así recomendaciones en base a los datos utilizados y las simulaciones generadas, mejorando así la seguridad y el bienestar de las comunidades costeras en Chile.

Objetivos

- **Objetivo general** Generar a partir de la información otorgada por instituciones como la ONEMI, una estructura, potencialmente una red, que modele la ciudad de Viña del Mar, con la finalidad de generar rutas de evacuaciones óptimas, teniendo en consideración densidad de población, zonas de peligro o de gran potencial, asignación de personal de soporte, de tal manera de trasladar la población hacia zonas objetivo como lo son los puntos de encuentro.
- **Objetivos específicos**
  - Recopilar información valiosa y descriptiva sobre le región, que permita definir espacios de búsqueda óptimos para los algoritmos.
  - Simular evacuaciones en la ciudad de Viña del Mar, utilizando la red asociada ésta.
  - Generar información relevante para entidades públicas, apoyando la toma de decisiones.

CAPÍTULO 2
DEFINICIÓN DEL PROBLEMA

En los últimos años han ocurrido dentro de nuestro país una gran cantidad de desastres naturales, tales como, terremotos, inundaciones, tsunamis, incendios forestales y hasta erupciones volcánicas. Se observa un aumento en la frecuencia, donde en los últimos 30 años han aumentado de 50 a 400 desastres naturales por año y se espera que estos sigan creciendo a una tasa similar[Kovacs, 2007]. Esto sin considerar los desastres generados por el humano, tales como, incendios, fallas estructurales y fuga de gases nocivos.

La ciudad de Viña del Mar no es ajena a esto, existiendo actualmente una gran área del plan de Viña del Mar catalogada por la ONEMI como zona de alto riesgo a la hora de un tsunami. Además de presentar diversos focos de incendios forestales, generando consecuencias fatales para la región, como se puede observar en este [Onemi, a].

Esta gran diversidad de fenómenos que pueden afectar a la ciudad requieren de una preparación, por lo que, es de suma importancia generar herramientas eficientes que logren identificar y generar rutas de evacuación dada una población afectada en zona de riesgo, movilizándola hacia diferentes puntos de encuentros, albergues o zonas seguras distribuidas a través de la ciudad. La categorización de zonas, tanto de peligro como seguras, se encuentran desarrolladas y estudiadas actualmente por entidades gubernamentales, como lo es la ONEMI en el caso de su [Onemi, b] y la municipalidad de Valparaíso con una ”Plan maestro

para la Gestión de Riesgos de Incendios en Valparaíso"[municipalidad de Valparaíso, 2018], generando así estudios geográficos, tanta como para identificar zonas seguras, como zonas de alto riesgo humano, ya sea por inhalación de gases tóxicos o el contacto con el incendio mismo, además de definir zonas con mayor potencialidad de incendio.

Actualmente se puede observar que al momento de prepararse para una catástrofe se suele informar a la comunidad a través de zonas seguras a la cual uno debe asistir por ciertos caminos estáticos, enseñando a la comunidad estas rutas a través de simulacros. Pero al momento de una catástrofe existe pánico e incertidumbre sobre el, por lo que si bien lo primero que se debe conocer es una ruta óptima para evacuar, entorno también se debe tener presente como responder a eventos secundarios que compliquen la evacuación, tales cómo, caminos dañados, corte de puentes y accidentes de tránsito.

Es por esto que al poseer un algoritmo, que tome en cuenta los diferentes parámetros se puede tener una gran ventaja frente a la toma de decisiones en situaciones adversas y de pánico, además de poder planear con anterioridad, la ocurrencia de diferentes sucesos, identificando falencias y oportunidades de mejoras, que podrían llegar a generar la fomentación de simulacros más variados y racionales con la utilización de estos algoritmos. Dada esta gran variedad se pueden clasificar los diferentes tipos de escenarios de manera simple como:

- Precautorio: En este tipo de evacuación se toma en consideración la estimación de tiempo de evacuación, con el tiempo de propagación de la catástrofe, por lo que el tiempo y los riegos potenciales son componentes claves para este tipo de evacuación
- Operaciones salvavidas: Este tipo de evacuaciones ocurren cuando no existe prevenciones necesarias antes de la emergencia a ocurrir, llevándonos mas a un problema de rescate de personas en áreas dañadas, despejar rutas, etc.

Por lo tanto, la importancia de desarrollar herramientas y modelos eficientes que permitan planificar y ejecutar evacuaciones óptimas en la ciudad de Viña del Mar. A través de la recopilación de información precisa y la simulación de escenarios de evacuación, se busca no solo mejorar la capacidad de respuesta ante desastres, sino también apoyar la toma de decisiones por parte de las autoridades. La utilización de algoritmos avanzados que consideren múltiples variables y posibles contingencias es esencial para aumentar la eficacia y seguridad de las evacuaciones, proporcionando a la comunidad rutas adaptables y racionales en momentos críticos.

En el siguiente capítulo se revisaran las diferentes metodologías utilizadas hasta la fecha que abarcan representaciones y soluciones de nuestro problema.

CAPÍTULO 3
Marco Conceptual

Este capítulo explora los conceptos y métodos fundamentales para planificar rutas de evacuación. Se destaca la importancia de los Sistemas de Información Geográfica (GIS) en el manejo de emergencias, como lo subraya GoodChild. Los GIS permiten un acceso rápido y preciso a la información, mejorando la respuesta ante desastres naturales.

Se revisan diferentes enfoques para la planificación de rutas de evacuación, incluyendo métodos de optimización de flujo en redes, modelos de simulación y modelos heurísticos. Cada uno de estos enfoques tiene sus propias ventajas y limitaciones, y se analizan en detalle para entender su aplicabilidad en distintos contextos de evacuación.

Además, se introduce el concepto de planificación de rutas de evacuación sin conflictos (CF-ERP) y el problema de evacuación de comunidades (CEP), explicando sus formulaciones y métodos de resolución. Finalmente, se discuten los modelos de simulación, tanto macroscópicos como microscópicos, y su capacidad para simular comportamientos individuales y dinámicos en situaciones de evacuación.

GoodChild definió la gran importancia de los *Geographic Information System* (GIS) [Goodchild, 2007] para el manejo de emergencias. Su enfoque hacia sistemas tecnológicos GIS y métodos para mejorar el tiempo de respuesta frente a tsnumais, incendios o terremotos. Lo convierte en unos de los primeros en fomentar el uso de las capacidades de GIS, tales como, rápido acceso a la data, precisión e interoperabilidad para apoyar la toma de decisiones. Para todo esto se requierede grandes volúmenes de data sobre diferentes aspectos en la región, por ejemplo, edificios, zonas comerciales, colegios y eventos, requiriendo heurísticas o metodologías computacionales para lograrlo. Teniendo en consideración desde un principio que las principal capacidad del sistema debe ser la rapidez, o sino estas terminaran siendo inútiles. En el problema *Evacuation Route Planning* (ERP), la estructura espacial es representada y analizada a través de una red, con restricción de capacidades positivas, tanto en los nodos, como las aristas, las cuales se definen con información del estado inicial de todas las personas a evacuar. Teniendo en consideración que existen nodos que representarán puntos de encuentros o zonas seguras, generando finalmente un conjunto de rutas origen-destino.

El problema **ERP** se puede definir utilizando $G = (V, E)$ un grafo dirigido donde V representa el conjunto de nodos (intersecciones y refugios) y E el conjunto de aristas (rutas) que conectan los nodos.

- $S \subseteq V$ es el conjunto de nodos de salida (puntos de evacuación).
- $D \subseteq V$ es el conjunto de nodos de destino (refugios seguros).

- c_{ij} es la capacidad de la arista $(i, j) \in E$.
- t_{ij} es el tiempo de viaje a través de la arista $(i, j) \in E$.
- x_{ij} es el flujo de evacuados a través de la arista $(i, j) \in E$.

En resumen se tiene que cada nodo, corresponde a una intersección de rutas o alguna infraestructura de cualquier índole, teniendo como atributos la capacidad de ese nodo y el número de personas presentes dentro de este. Así mismo cada arista es definida por un camino, calle, carretera, con su máxima capacidad y costo de la ruta en si. A continuación se representa una simple red para (ERP).

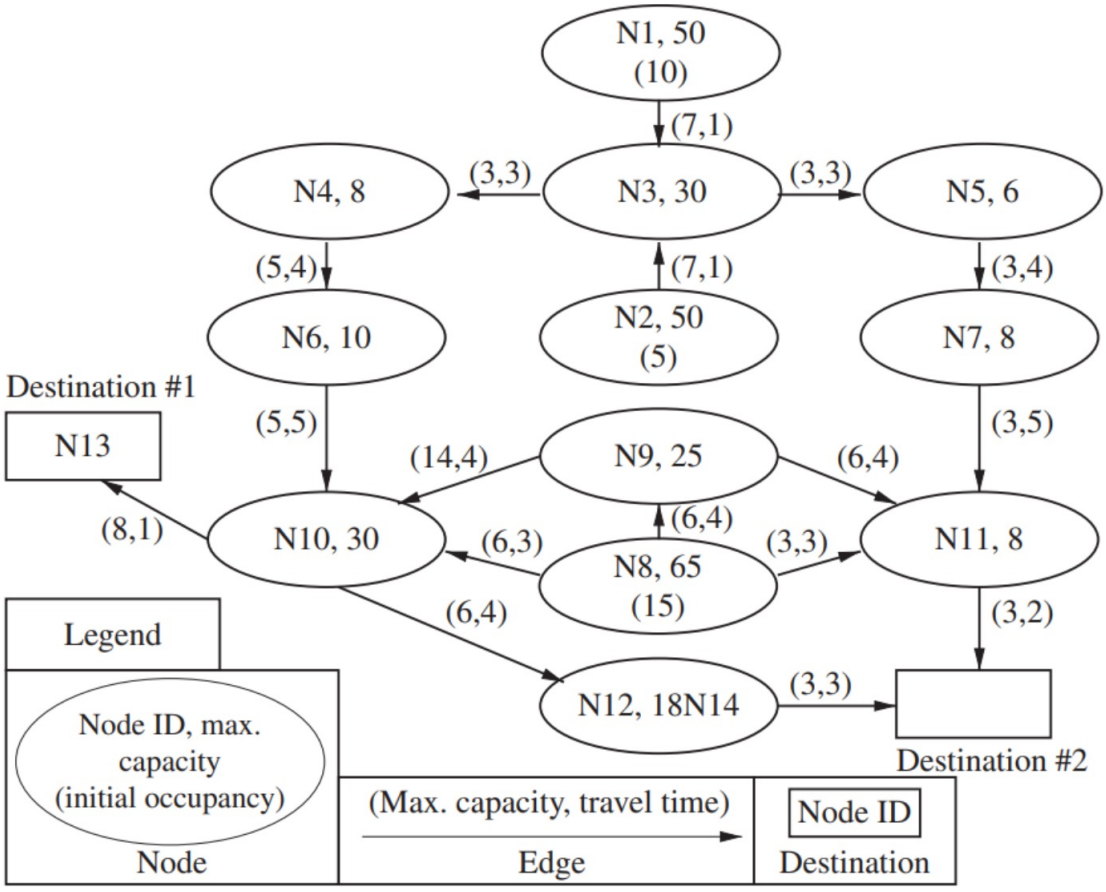


Figura 1: Red simple, ERP

Actualmente existe una gran cantidad de investigación realizada en *Evacuation Route Planning*(ERP), de las cuales se suelen dividir en 3 grandes categorías:[Herschelman *et al.*, 2021]

- Modelos Heurísticos, en busca de soluciones incompletas con un menor costo computacional. [Lu *et al.*, 2005]

- *Network Flow Optimization Methods*, utilizando técnicas de optimización como *max-flow* o *min-cost Flow* [Dong *et al.*, 2019] [Hamacher y Tjandra, 2001]
- *Simulation Model*, el cual modela las rutas de evacuación como movimientos individuales o como un problema de asignación de tráfico.

3.1. *Modelos Heurísticos*

Los modelos Heurísticos pueden ser utilizados para definir restricciones de capacidad para *Evacuation Route Planning* y lograr encontrar soluciones cercanas al óptimo teniendo en consideración una reducción del costo computacional, el cual se basa en movilizar a través de la red de transporte un cierta cantidad de pesos utilizando las capacidad y costos de los arcos. [Yang *et al.*, 2012] [Kim *et al.*, 2007][Lu *et al.*, 2007]. Aunque lamentablemente la experiencia ha demostrado que de estos enfoques no poseen restricciones de libre-conflicto, por lo que se ignoran muchas las interacciones entre la red, fomentando escenarios adversos, tales como, congestión vehicular o conflictos de movimiento en si.

3.2. *Network Flow Optimization methods*

La principal idea de estos modelos es representar un lugar, y los atributos de la zona a través de una red G, para luego modelar la evacuación a través del tiempo en una red dinámica G_T, siendo una versión extendida de G, generando así que el flujo de la red corresponda a la evacuación.[Dyn, 2001]

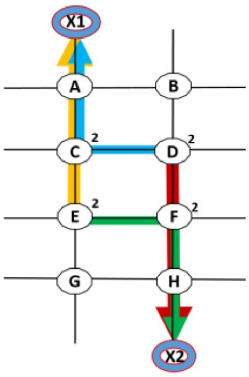
3.2.1. *Conflict Free Evacuation Route Planning* (CF-ERP)

Se crean algoritmos del tipo *Conflict Free Evacuation Route Planning*(CFERP) [Herschelman y Yang, 2019], aumentando el desarrollo del ERP, definiendo 2 conceptos claves:

- **Conflicto:** Dadas dos rutas diferentes, un conflicto es definido cuando dos rutas se encuentran en una intersección (Por ejemplo un nodo) en el mismo tiempo, llevando así a todas las consecuencias anteriormente mencionadas.
- ***First-In First-Out:*** Un vehículo o entidad a evacuar no puede adelantar a otras, o sea todas van a una misma velocidad, generando asi agentes homogéneos entre ellos, simplificando el problema.

La formulación de CF-ERP, es similar a las mencionadas anteriormente, donde se tiene $CF-ERP(N, E, C, T, n, k)$

- Un conjunto de nodos N, y un conjunto de E vértices.
- un conjunto de enteros positivos asociadas a la restricción de capacidad $C : E \to Z$
- un conjunto de enteros no negativos asociado al tiempo de cada vértice $T : E \to R$
- El número total de evacuados y su posición inicial.
- Parámetro asociado a la resolución de conflicto. Donde se tiene un porcentaje de nodos para los cuales los conflictos serán resueltos, entre toda la red y durante todo el tiempo de evacuación.



Source	No. Of Evacuees	Route Node (Time)	Arrival Time
C	1	C(1)-A(2)-X1(3)	3
	1	C(2)-A(3)-X1(4)	4
F	1	F(1)-H(2)-X2(3)	3
	1	F(2)-H(2)-X2(4)	4
D	1	D(2)-C(3)-A(4)-X1(5)	5
	1	D(2)-F(3)-H(4)-X2(5)	5
E	1	E(2)-C(4)-A(5)-X1(6)	6
	1	E(2)-F(4)-H(5)-X2(6)	6

Figura 2: Ejemplo CF-ERP [Herschelman y Yang, 2019]

Teniendo como salida, un conjunto de rutas origen-destino y el plan para los evacuados en cada ruta. Los autores proponen diferentes metodologías para CF-ERP, como por ejemplo, *Suffix-Tree Conflict Minimazation*(STCM), identificando los conflictos entre peatones y reduciéndolos. Por ejemplo el caso en la figura 3 se puede observar una representación de las rutas y a su lado una tabla con las rutas a tomar por cada evacuado y su tiempo de llegada respectivo. Donde la planificación otorgada para cada individuo es a través de la pareja Nodo y tiempo, otorgándo la información necesaria para saber donde se encuentra cada evacuado en diferentes tiempos. Pero es aún mas importante rescatar que los conflictos permitidos ocurren en los nodos C y F, dado que estos reciben flujo desde D y E.

3.2.2. *Community Evacuation Problem* (CEP)

El problema de evacuación de comunidad es trasladar a todo quien necesite ser traslado, independiente de su localidad, a un lugar seguro lo antes posible, teniendo en consideración el número de personas a evacuar en cada nodo o comunidad, la restricción de capacidad de cada ruta, la capacidad de cada nodo seguro, y la relación no lineal entre tiempo de evacuación y número de personas evacuadas en cada ruta, debido a que el movimiento de una multitud se ve reducido por tener una mayor densidad .[Liu *et al.*, 2022] Los conjuntos $I = 1, 2, ..., m$

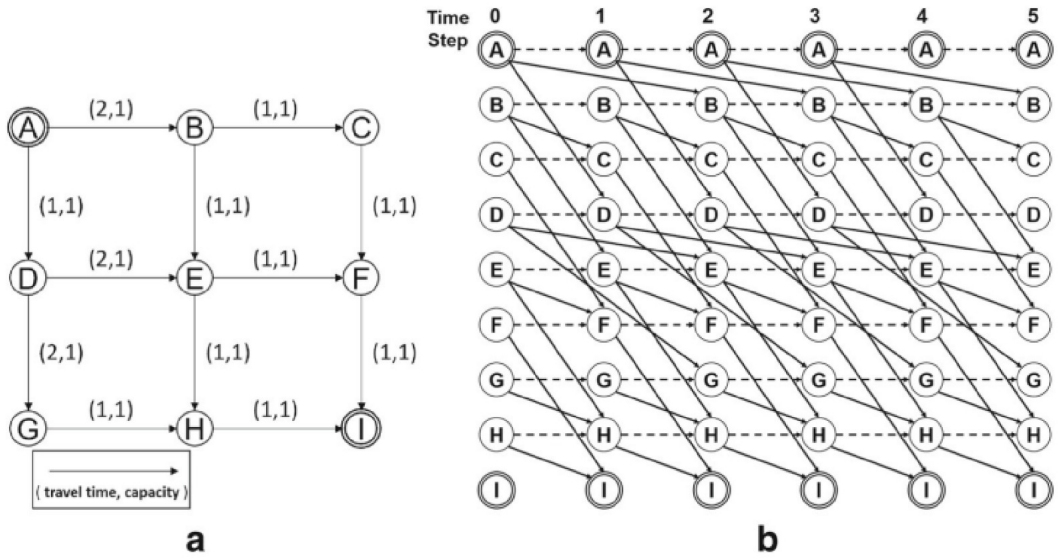


Figura 3: STN, expansión temporal

y $J = 1, 2, ..., n$, representan el conjunto de comunidades y puntos de encuentros respectivamente, b_i, representa los números de personas pertenecientes a esa comunidad i, c_j, representa la capacidad máxima de la comunidad j, P_{ij} representa la ruta entre la comunidad i, con el punto de encuentro j. x_{ij} es la variable de decisión y representa el número de peatones asignados en la ruta p_{ij}. Para luego definir el tiempo de evacuación de esta ruta como una función convexa, representada por $g_{ij}(x_{ij})$. Ya que diferentes cantidades de peatones en una ruta corresponderá a diferentes tiempo de evacuación. Dado esto se puede representa el modelo:

$$\min_{g} \max_{i,j} \{g_{ij} x_{ij}\} \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = b_i, i = 1, 2, ...m; \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \leq c_j, j = 1, 2, ...n; \tag{3}$$

$$x_{i,j} \geq 0, x \in \mathcal{Z} \tag{4}$$

Se asume que no existe influencia mútua entre diferentes caminos, es decir, estas rutas siempre estarán conectadas a través de nodos.

3.2.3. *Spatiotemporal Network* (STN)

Las rutas de evacuación pueden ser vistas como un flujo con restricciones de capacidad, viéndose reflejadas en una STN. Las STN pueden ser representadas por un grafo con atributos

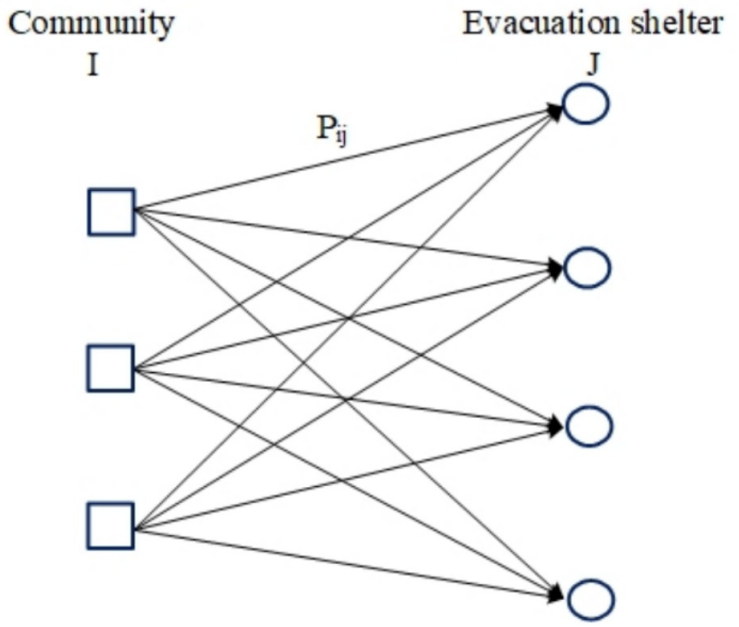


Figura 4: Diagrama CEP

temporales, compuesta por nodos(N) y vértices (E), y tiempo. Cada nodo representa un espacio geográfico, cada arista la conección entre estos nodos.

Teniendo en consideración la Figura 3, se observa un caso de 9 nodos y 12 aristas, cada arista posee un tiempo de viaje y una restricción de capacidad. Asumiendo que el nodo *A* es un punto de inicio y el nodo *I* un nodo destino. El gráfico espacial puede ser modelado por un *Time Expanded Graph*(TEG), el cual es una representación de los nodos en cada linea de tiempo. Por ejemplo, cada nodo temporal en TEG, esta asociado a un tiempo $(A(0), A(1)...A(5))$. donde cada iteración debe mantener las restricciones antes mencionadas según cada nodo y arista.

La limitación que poseen estos métodos es el gran uso de memoria y, por ende, tamaño de la STN, aumenta cuando el tiempo óptimo de evacuación aumenta, limitando a su vez modelos con una red de gran cantidad de nodos.

3.3. *Simulation Models*

Los modelos de simulación de evacuaciones, son clasificadas en dos grandes áreas, microscópica y macroscópica. Los modelos macroscópicos considerara la población como una gran entidad homogénea, mientras que los modelos microscópicos analizan a la población como entidades dinámicas capaces de interactuar entre ellas y poseer comportamientos individuales. Los modelos microscópicos pueden formularse de manera discreta, como *Cellular Automata)*(CA), los cuales pueden ser utilizados para pequeñas a medianas poblaciones, o

continuo, como *Agent based Models* (ABMs)[Zheng *et al.*, 2009].

Los modelos macroscópicos dada su naturaleza fallan en identificar comportamientos sociales de cada individuo, ignorando la toma de decisión personal de los individuos, limitando estos modelos a situaciones donde el contexto es simple, o sea, con bajos obstáculos dentro de las rutas. Los modelos macroscópicos enfocados en dinámica de fluidos, pueden llegar a ser muy complejos de implementar, debido a sus ecuaciones diferenciales no lineales y la difícil definición de hipótesis que requieren éstas. Por otro lado, los modelos microscópicos si bien logran incorporar comportamientos individuales, éstos en la práctica fallan en predecir el comportamiento humano frente a la incertidumbre de una evacuación. La ventaja de *ABMs* es que al definir las interacciones a una escala mínima, se pueden observar diversos e inesperados fenómenos en escalas macroscópicas. Aunque las ABMs pueden ser computacionalmente altamente costosas, poseen la ventaja de implementar comportamientos únicos de entidades heterogéneas, una característica interesante y valiosa que otros modelos no poseen, dándo libertades de poder modelar la ciudad de Viña del Mar con mayor precisión y fidelidad de la realidad. Para simulaciones de tan gran escala, el tiempo de computo es un factor decisivo y problemático a la hora de sucesos que requieran de una respuesta pronta, como lo son los incendios y maremotos, donde además es necesario prever diferentes escenarios.[Castro *et al.*, 2019]

3.3.1. ABMs

Los ABM de gran escala, están asociadas comúnmente a una gran costo computacional. Disminuir este tiempo de ejecución, es un desafío complejo, al momento de relacionarse con la capacidad de operaciones y aplicaciones, en contextos con restricción de tiempo como lo es un proceso de evacuación. Existen 3 diferentes opciones frente a este problema. Primero y la más simple es invertir en máquinas con alto nivel de computo, agregar ABMs con índoles mas generales o sea macroscópica y finalmente, simplificación de modelo.

Existe una gran variedad de estudios y simulaciones utilizando ABM, teniendo en consideración el modelado autónomo de agentes, existiendo diferentes software, tales como, MATSim, EXODUS [Veeraswamy *et al.*, 2018] y *Pathfinder*. Todos estos poseen similitudes al momento de modelar el movimiento de agentes, por ejemplo, algoritmos*shortest-path* para planeamiento de rutas y *social forced-based*, para evitar colisiones entre agentes. Aunque igual se han desarrollado software donde se observa que los agentes poseen 3 tipos de comportamientos, locomoción (camina, corre, para, giro), dirección (buscar, seguir, evitar colisiones) y social (competitivo, haciendo fila, *"pastoreo"*). [Pan *et al.*, 2007]

3.3.2. *Agent Based Evacuation Model*(ABEM)

Mordvintev [Mordvintsev *et al.*, 2014], desarrollo un modelo ABEM, acoplado con la simulación de mecánica de fluidos, donde el área inundada se propaga dinámicamente mientras los peatones se dirigen hacia puntos de encuentros previamente dispuestos y localizados, utilizando algoritmos de planeación global. Aplicando diferentes iteraciones con el fin de actualizar el plan global, considerando como el fluido se propaga sobre la ciudad. Para evitar colisiones con obstáculos para los peatones se utiliza *Optimal Reciprocal Collision Avoidance*(ORCA), además de definir simulaciones de peatones bajo diferentes probabilidades, teniendo en cuenta la características especificas, como por ejemplo, iluminación de las calles, presencia de personal especializado en catástrofes, altura de edificios o entorno en general.

El capítulo anterior examina los conceptos y métodos esenciales para la planificación de rutas de evacuación, destacando el papel crucial de los Sistemas de Información Geográfica (GIS) en la gestión de emergencias, según GoodChild. Se analizan diferentes enfoques, incluidos métodos de optimización de flujo en redes, modelos de simulación y heurísticos, junto con sus ventajas y limitaciones. Se introduce la planificación de rutas de evacuación sin conflictos (CF-ERP) y el problema de evacuación de comunidades (CEP), detallando sus formulaciones y métodos de resolución. También se exploran modelos de simulación macroscópicos y microscópicos, resaltando su capacidad para simular comportamientos individuales y colectivos en evacuaciones.

CAPÍTULO 4
PROPUESTA DE SOLUCIÓN

La eficacia de los algoritmos de evacuación en una primera instancian dependen en gran medida de la precisión con la que se representa la red de transporte sobre la cual operan. Contar con una representación fiel de la realidad en la infraestructura de transporte se vuelve fundamental para salvar vidas y minimizar el caos en situaciones críticas. Es por esto que el primer paso, el cual definirá los resultados en mayor medida, será crear redes de transporte con la mayor fidelidad posible para garantizar la efectividad y fiabilidad de los algoritmos de evacuación.

Dado este contexto, la biblioteca *OSMnx* (Open Street Map networkx) ofrece una solución robusta para la creación de modelos detallados y precisos de redes de transporte basados en datos de *OpenStreetMap* (OSM), los cuales son actualizados constantemente y cuentan con una gran cooperación entre sus colaboradores, asegurando así la fidelidad y validación de los datos. Por lo tanto, utilizar *OSMnx* se presenta como la mejor opción para generar la red de transporte, destacando los atributos proporcionados por esta data y los beneficios que ofrece frente a otras metodologías de modelado.

Los conjuntos de datos descargados a través de *OSMnx* abarcan una amplia gama de información sobre la infraestructura vial y la red de transporte de una determinada área geográfica, lo cual puede ser adaptado a las necesidades específicas del usuario. Estos datos incluyen, entre otros elementos, la ubicación geográfica precisa de nodos, utilizando coordenadas de latitud y longitud con 5 decimales de precisión, los cuales representan intersecciones, cruces y otros puntos de interés en la red vial. Además, los datos contienen atributos detallados sobre las conexiones entre estos nodos, representando asi nuestros arcos a utilizar, que pueden ser carreteras, calles, autopistas, pasajes peatonales, ciclovías entre otros tipos de vías, generando información útil para definir la capacidad de transporte caracterizando la gran variedad de calles o arcos presentes en ella. Estos arcos también pueden incluir la subclasificación de la carretera o arco (por ejemplo, autovía, carretera secundaria, calle residencial), el sentido de circulación unidireccional o bidireccional, la velocidad máxima permitida, el número de carriles, y el largo de ella con una precisión en centímetros y cualquier restricción o características especiales asociadas con la vía, lo que permite una categorización específica según sea nuestro interés, como mencionaremos posteriormente. En conjunto, estos datos proporcionan una representación exhaustiva y detallada de la infraestructura vial de una región específica, lo que permite una modelización precisa y una planificación efectiva de la red de transporte, al igual que una libertad creativa para nuestro análisis o creación de variables para una futura representación visual, como la imagen a continuación representando los arcos que constituirán nuestra red de transporte, obtendremos un grafo dirigido de 15604 nodos y 36773 arcos, presente en la figura 5

La utilización de datos proporcionados por *OSMnx* ofrece varios beneficios significativos en

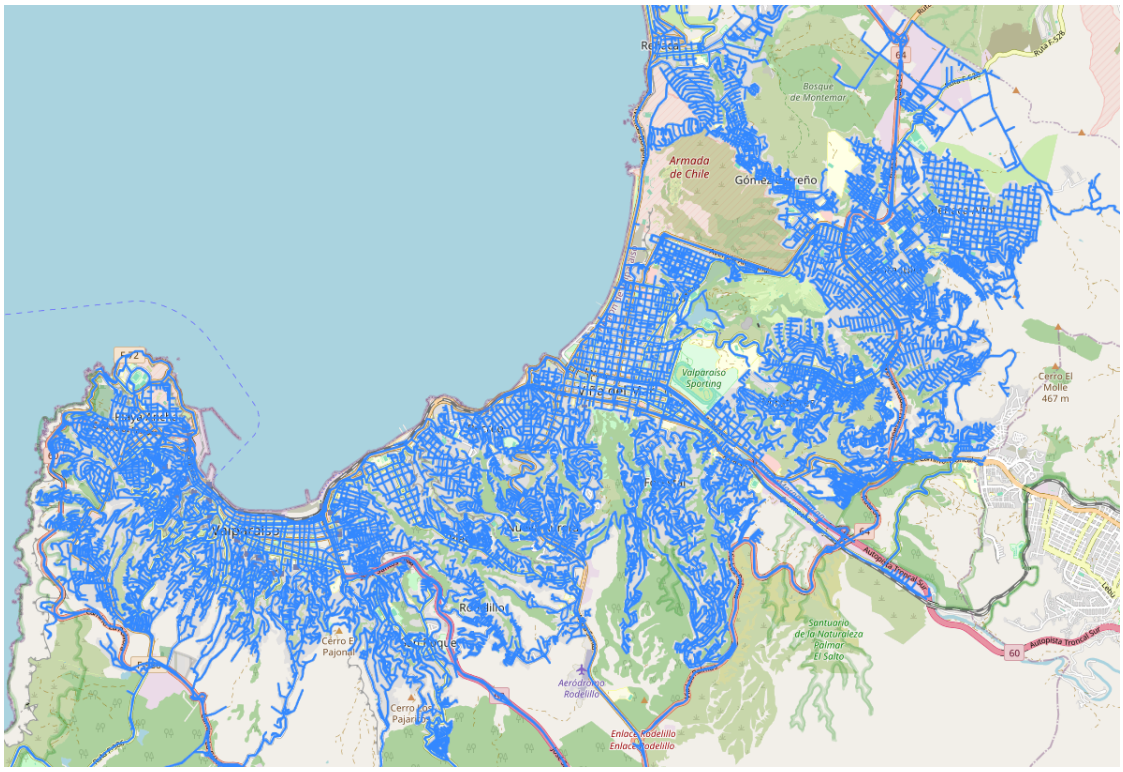


Figura 5: Red de transporte
Fuente: Elaboración propia

comparación con otros métodos de modelado de redes de transporte. En primer lugar, la información proporcionada por *OSM* es de acceso abierto y gratuito, lo que permite una fácil disponibilidad para una amplia gama de usuarios, desde investigadores académicos hasta planificadores urbanos y responsables de la gestión de emergencias. Además, la comunidad de *OpenStreetMap* se encarga constantemente de mantener y actualizar los datos, lo que garantiza que la información esté actualizada y refleje con precisión los cambios en la infraestructura vial. Esto es crucial para la planificación de evacuaciones y la gestión de crisis, donde la precisión y la actualización de los datos son fundamentales para una respuesta efectiva. En resumen, el uso de datos proporcionados por *OSMnx* ofrece una solución conveniente, precisa y actualizada para la representación de redes de transporte, lo que permite una planificación de evacuación efectiva y una gestión eficiente de emergencias.

Teniendo los datos necesarios para generar la red de transporte y los atributos que describen cada conexión, y abarcando el caso específico de la región de Valparaíso, principalmente las comunas de Viña del Mar y Valparaíso, se observa en el dataset dentro y su gran variedad de atributos calificativos se consideran únicamente los atributos presentes de forma constante, los cuales son, largo de calle con precisión de centímetros, clase asociada a la calle, dependiendo si se desea extraer todos los caminos, ya sean caminos aptos para peatón, vehículos y/o bicicletas o únicamente manejables, en el caso de estudio solo se utiliza la red de transporte apta para vehículos, obteniendo así las siguientes clases dentro de los arcos o calles.

Tabla 1: Highway Capacity

Tipo de Calle	**Factor de Capacidad**
motorway	1.10
motorway_link, trunk, trunk_link	1.00
primary, primary_link, secondary, tertiary, tertiary_link	0.95
secondary_link	0.90
residential, living_street	0.85
track, service	0.70
path, unclassified,	0.60

Como se observa en la tabla 1 se genera un factor de capacidad asociado a cada clase con el fin de representar la variabilidad de cada una de ellas y su cualidad de abarcar diferentes cantidades de autos independientemente del largo de ella. Este factor es estipulado en base a la Wiki de **OSM**, enfocada principalmente en los anchos de la calle.

- **motorway:**Calle principal con al menos dos lineas y presencia de bermas, capaces de abarcar una mayor cantidad de autos dada la situación de emergencia, o vehículos que poseen mayor agilidad como las motocicletas.
- **motorway-link:** Siguiente calle que no alcanza a ser calle principal o de unión de diferentes comunas, principalmente avenidas.
- **primary,primary-link:** Calles principales presentes en la ciudad, participes dentro de la red vial común.
- **residential, living-street:** Calles donde los peatones poseen prioridad sobre los vehículos en una situación común.
- **track,service:**Calles presentes en zonas rurales, de bajo trafico.
- **path,unclassified:** Calles hábiles para transito de vehículos pero con características informales, o sin pavimentar.

Considerando los diferentes tipos de datos se genera una función de costo asociado a cruzar un arco en específico, al igual que se otorga un valor capacidad asociado a la cantidad de vehículos que pueden abarcar sin generar un atascamiento.

- **distancia:** Largo de calle con precisión de centímetros, por ejemplo $198,25[metros]$, pero para no utilizar números flotantes y aumentar aun mas los requerimientos computacionales, al igual que dificultar las operaciones de obtener rutas de menor costo dentro de la red, se multiplica la distancia por 10, transformando cada unidad de la distancia a $10[cm]$, otorgando la capacidad de ser utilizada como *int*, y manteniendo una precisión aceptable para la medición de calles.

- **capacidad:** Distancia * Factor de Capacidad * Número de lineas.
- **A salvo:** Se define para cada nodo según coordenada latitud y longitud si es que esta presente en la zona de peligro.

4.1. Zona de peligro

La presente memoria se enfoca en la evacuación del caso particular de tsunamis, por lo que se utilizan los datos otorgados por la SENAPRED (El Servicio Nacional de Prevención y Respuesta ante Desastres) para eventos para olas de 30 metros, a disposición en en el visor para una mayor exploración, estos dato son Considerados Cota30, y se enfocan en la altitud de las zonas, se observa específicamente que el centro de Viña del Mar esta bajo la cota30 al igual que el centro de Valparaíso.

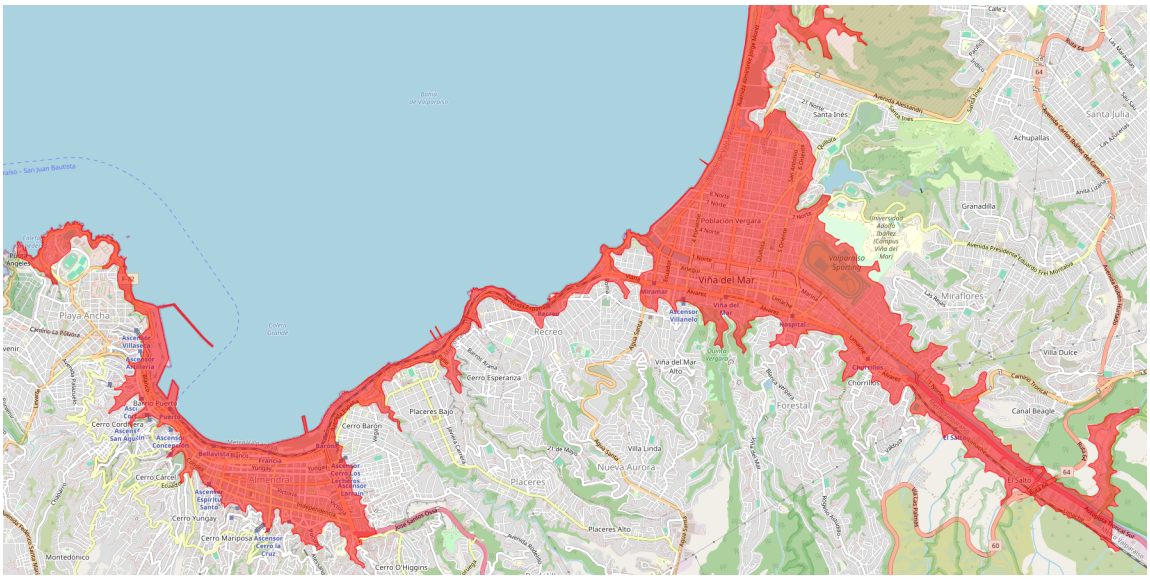


Figura 6: Zona de Peligro
Fuente: Elaboración propia

Cabe destacar que al considerar para el caso de estudio los nodos y arcos que estén en peligro o a salvo, se generan para cada uno de los casos, 2 nodos *dummy*, donde un nodo se encuentra conectado a todos los nodos a salvo, mientras que todos los nodos en peligro conectan a otro nodo *dummy*, esto es generado con el fin de facilitar ciertas tareas y algoritmos que utilizaremos a lo largo del desarrollo de la evacuación. En una primera instancia se experimento con algoritmos de maximización de flujo de grafos, teniendo en cuenta las capacidad y costos de cada arco, donde se genera un flujo desde estos dos nodos *dummy* como origen y destino para el calculo de flujo de cada arista, pero esto no termina siendo útil para el caso de estudio, debido a que no se toma en consideración la distribución de población. Todos los nuevos arcos que conectaran estos nodos a los nodos *dummy* serán asignados con una capacidad en base a la sumatorio de cada nodo en específico. En el caso de los arcos asociados

de los nodos en peligro, todas las conexiones donde el nodo en peligro que busca la nueva conexión sea el origen y en el caso de los nodos a salvo donde el nodo sea el destino. El nodo *dummy* a salvo lo llamaremos ***node_dummy_safe*** y el en peligro ***node_dummy_danger***.

Por otro lado estos nodos también serán útiles para generar las rutas de menor costo desde los nodos en peligro a uno a salvo sin tener la necesidad de probar en cada una de las opciones, sino que todas las rutas serán dirigidas hacia el nodo *dummy* que conecta a todos los nodos a salvo, con la salvedad de que el costo de este arco es 0 y su capacidad es teóricamente infinita.

4.2. Estimación de población

Al abordar el problema de estimar la distribución de los habitantes en un área urbana, la de por si es un problema complejo ya que esta varia y se distribuye en diferentes formas teniendo en consideraciones variables como la hora del día, época del año o porcentaje de población flotante, otorgándonos así una importancia de comprender cómo se utilizan las variadas herramientas y métodos para asignar de manera efectiva la población a ubicaciones específicas dentro de esa área. Siendo este problema útil de resolver o estipular para una amplia gama de aplicaciones, desde la planificación urbana, la gestión de emergencias, evaluación de riesgos y tendencias de ubicación de la población.

Uno de los enfoques comunes para estimar la distribución de población es utilizar datos geoespaciales y técnicas de análisis espacial para asignar población a nodos específicos en una red de transporte, como se estipula anteriormente. Este enfoque se basa en la premisa de que la distribución de la población urbana está estrechamente relacionada con la infraestructura de transporte y la accesibilidad a diferentes partes de la ciudad, y más aun cuando necesitamos asignar a nuestra red de transporte su población, generando así un lazo entre la infraestructura comercial y demográfica hacia la red de transporte.

En este contexto, una de las herramientas que se emplea es el uso de diagramas de Voronoi, también conocidos como polígonos de Thiessen. Estos diagramas dividen el espacio en regiones basadas en la proximidad a puntos de referencia específicos, lo que proporciona una representación visual de las áreas de influencia de cada uno de estos puntos, abarcando así el área de un polígono para cada nodo de nuestra red. Teniendo en consideración que este polígono sera capaz de representar según su área y forma la jerarquía de cada nodo, ya que si estos nodos se encuentran cercanos entre si, el polígono asociado tendrá una menor área, y esta área además sera capaz de crecer hacia las direcciones mas vacías, teniendo como principal características que si se posiciona un nuevo nodo, basta con observar dentro de que polígono esta contenido para saber cual es su nodo mas cercano.

Una vez elegida esta implementación se genera en base a los nodos y su posición longitud y latitud, para asignar el conjunto de puntos asociados a los nodos en un plano euclidiano, con los cuales se obtienen los polígonos asociados a cada uno de ellos y a su vez el área que

encapsula cada nodo, con las cuadras de población.

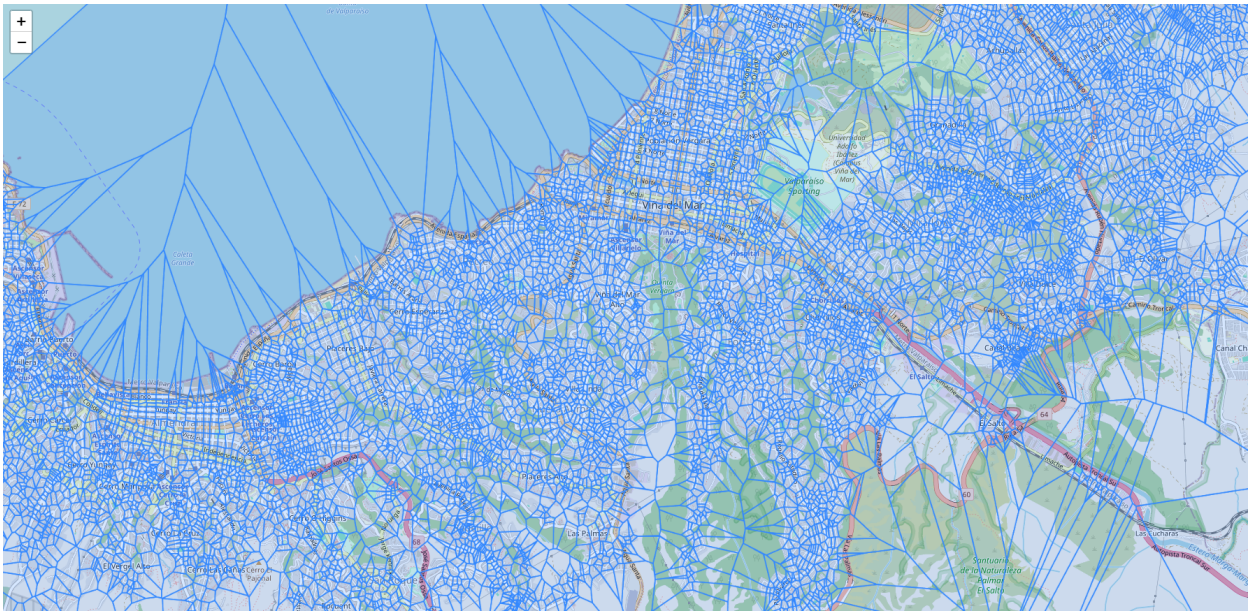


Figura 7: Polígonos de Voronoi
Fuente: Elaboración propia

En resumen, la estimación de la distribución de población en un área urbana implica el uso de datos geoespaciales, técnicas de análisis espacial y herramientas como los diagramas de Voronoi para asignar de manera efectiva la población a ubicaciones específicas dentro de esa área.

En este sentido, los polígonos de Voronoi se utilizan para asignar población a los nodos de una red de transporte. Este proceso implica la intersección de los polígonos de Voronoi asociados a cada nodo de la red, con las cuadras que conforman la infraestructura de la ciudad. La intersección resultante proporciona un área común entre el polígono de Voronoi del nodo y las cuadras de la ciudad, que se utiliza para calcular la proporción de población asignada a ese nodo en particular.

4.2.1. Metodología para la Generación de Polígonos de Voronoi

Sea $P = \{p_1, p_2, ..., p_n\}$ el conjunto de puntos cartesianos bidimensionales que representan las ubicaciones de intersección de calles, osea los nodos de la red de transporte, y sea B el límite del área de estudio, opcional pero recomendado para definir polígonos de Voronoi externos, ya que estos pueden generar polígonos en los extremos de nuestra red de gran tamaño.

Algoritmo 1 Generar Polígonos de Voronoi

Entrada: Conjunto de puntos P en el plano
Salida: Lista de polígonos de Voronoi $poligonos\_voronoi$
$bordes\_espacio \leftarrow CalcularBordesEspacio(P)$
for all punto p en P **do**
  Inicializar $bordes\_p$ como una lista vacía
  for all otro_punto q en P (distinto de p) **do**
    $linea\_bisectriz \leftarrow CalcularLineaBisectriz(p, q)$
    $puntos\_interseccion \leftarrow CalcularPuntosInterseccion(linea\_bisectriz, bordes\_espacio)$
    Agregar $puntos\_interseccion$ a $bordes\_p$
  end for
  $poligono\_p \leftarrow CalcularPoligonoVoronoi(bordes\_p)$
  Agregar $poligono\_p$ a $poligonos\_voronoi$
end for
Devolver $poligonos\_voronoi$

4.2.2. Cálculo del Diagrama de Voronoi:

Aplicar la triangulación de Delaunay a los puntos de entrada P para obtener el grafo de triangulación de Delaunay DT. Para cada triángulo t en la triangulación de Delaunay DT:

- Calcular el circuncentro c (vértice de Voronoi) del triángulo t usando la fórmula:

$$c = \left(\frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3} \right)$$

  donde (x_1, y_1), (x_2, y_2) y (x_3, y_3) son las coordenadas de los vértices del triángulo t.
- Agregar c al conjunto de vértices de Voronoi V.
- Para cada borde e del triángulo t:
  - Calcular el punto medio m del borde e.
  - Crear un borde de Voronoi ve conectando el vértice de Voronoi c con el punto medio m del borde e.
  - Agregar ve al conjunto de bordes de Voronoi E.

4.2.3. Manejo de Límites:

Si se proporciona un límite B:

- Recortar los polígonos de Voronoi al límite B para asegurar que no se extiendan más allá del área de estudio, esto sirve para simplificar la estructura de los polígonos.

Una vez generado los polígonos y asociado a cada uno de ellos su nodo, se utiliza la base de datos del Censo 2017 para obtener los datos de cada cuadra estipulada por ellos, este dataset presenta las cuadras como polígonos y dentro de cada cuadra se obtienen las siguientes columnas:

- **Entero:** OBJECTID, CUT, COD_DISTRI, COD_LOCALI, COD_ENTIDA, COD_CATEGO, **PERSONAS**, VIVIENDAS, HOMBRES, MUJERES, DE_0_A_5_A, DE_6_A_14_, DE_15_A_64, DE_65_MAS_, INMIGRANTE, PUEBLOS_OR, VIV_OCUPAD, TOTAL_VIVI, TOT_HOGARE, CASA, DEPTO, VIV_INDIGE, CONVENTILL, MEDIAGUA, MOVIL, OTRA_VIV_P, VIV_HORMIG, VIV_ALBANI, VIV_TABIQU, VIV_TABI_1, VIV_ADOBE, VIV_PARED, VIV_TEJAS, VIV_LOSA_H, VIV_PLANCH, VIV_FONOLI, VIV_PAJA, VIV_TECHO_, VIV_SIN_TE, VIV_PARQUE, VIV_RADIER, VIV_BALDOS, VIV_CAPA_C, VIV_PISO_T, IND_MAT_AC, IND_MAT_RE, IND_MAT_IR, VIV_AGUA_R, VIV_AGUA_P, VIV_AGUA_C, VIV_AGUA_1
- **Decimal:** MANZENT_I, SHAPE_Leng, SHAPE_Area, PERSONAS_LOG
- **Texto: REGIÓN**, PROVINCIA, **COMUNA**, DISTRITO, CATEGORÍA
- **Geometría: Polígono**

De las cuales solo se utilizaran la cantidad de personas, las comunas de Viña del Mar y Valparaíso y los polígonos de las cuadras, para luego iterar y calcular las intersecciones de cada polígono de voronoi con los polígonos asociadas a las cuadras, teniendo en consideración el porcentaje de la cuadra abarcada por el polígono de voronoi, ya que si esto fuera al contrario no se logra extraer todas las personas de la cuadra ya que los polígonos de voronoi poseen una mayor área que las cuadras y algunos ocupan gran parte de su área en el mar o lugares excluyente de las comunas a considerar, algo a tener en cuenta es que las cuadras no se sobreponen entre si y tampoco los polígonos de voronoi.

Cabe destacar que la principal necesidad de utilizar una escala logarítmica para representar la escala de color de las cuadras es que están varían en gran medida ya sea por la cantidad de área que abarcan, como también por la simple naturaleza de las construcción ya que cuadras con departamento aumentan en gran medida el número de personas en ellos, un ejemplo específico útil para explicar esto son las cuatro torres conocidas de la universidad Federico Santa María, donde se distribuye en las 4 intersecciones que presenta esta cuadra la población total de esta cuadra debido a la naturaleza cuadrada de las calles en esa zona, algo que se observa predominantemente en el plan o centro de ambas comunas, pero aun así es capaz de en cuadras con formas mas azarosas o caóticas asignar a los nodos mas cercanos la cuadra, y ayuda a ejemplificar la naturaleza del algoritmo de voronoi utilizado para transportar la población a la red de transporte.



Figura 8: Cuadras escala logarítmica de población
Fuente: Elaboración propia

Aun así se observa que este método también posee debilidades, ya que es sabido que esta región posee construcciones que cubren los cerros de forma improvisada y sin una organización definida o aparente, teniendo en consideración las complejas y variadas formas que un cerro puede llegar a tener, esto genera que ciertas cuadras solamente tengan salida de vehículos y peatonal hacia una dirección, como por ejemplo la población de la universidad se enlazara hacia avenida España, sin necesariamente poseer una salida de auto, esto puede ocurrir en diferentes cuadras dentro de los cerros. Aunque por otro lado la área de estudio para la evacuación es en gran medida dentro del plan o centro de las comunas, por lo que este error no prevalece dentro del análisis.

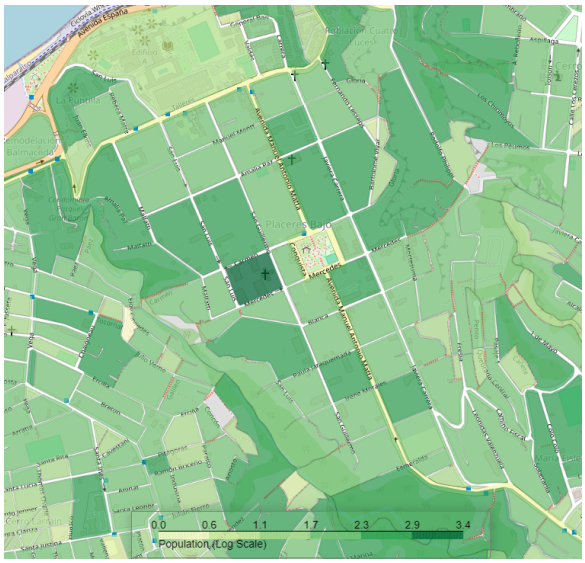


Figura 9: Voronoi según Población, Zoom Placeres
Fuente: Elaboración propia

Algoritmo 2 Calcular Porcentaje de Intersección

Input: Nodo asociado al polígono de Voronoi $nodo\_id\_a\_voronoi$, GeoDataFrame Censo $gdf\_polgonos\_censo\_Vina\_Valpo$
Output: Lista de intersecciones $Porcentaje\_de\_intersecciones$
$Porcentaje\_de\_intersecciones \leftarrow$ []
for all nodo_id, polígono_voronoi **in** $nodo\_id\_a\_voronoi$ **do**
  for all cuadra_info **in** $gdf\_polgonos\_censo\_Vina\_Valpo$ **do**
    $personas \leftarrow cuadra\_info['PERSONAS']$
    $polgono\_cuadra \leftarrow cuadra\_info['geometry']$
    $porcentaje \leftarrow$ porcentaje_intersección$(polgono_cuadra, polgono_voronoi)$
    if $porcentaje > 0$ **then**
      $Porcentaje\_de\_intersecciones$.append$([personas,$ node_id$, porcentaje])$
    end if
  end for
end for

Una vez obtenidas todas las intersección se poblan los nodos según los datos correspondientes y se genera el siguiente mapa con los polígonos de voronoi y su población asignada, en escala logarítmica igualmente. Teniendo en cuenta que dentro de la comuna de Viña del Mar existen 332875 personas y en la comuna de Valparaíso 294207 personas, censadas el año 2017, siendo la distribución de población considerada que todos los habitantes se encuentran en la dirección censada, o sea, sus domicilios.



Figura 10: Voronoi según Población (Escala logarítmica)
Fuente: Elaboración propia

4.3. Parque vehicular Valparaíso y Viña del Mar:

En esta sección se expresan las metodologías que representaran la distribución y magnitud de vehículos de la área de estudio, con el fin de generar la evacuación enfocada en rutas para estos vehículos, se enfoca en la distribución de población para enlazar aquella a sus respectivos medios de transportes.

Primeramente para obtener la cantidad de vehículos se utilizan los datos del parque vehicular generada por el Instituto Nacional de Estadística (INE), en su publicación de resultados del año 2022, estos estipulan la cantidad vehículos tanto particulares como comerciales, asociando a cada un espectro de categorías de diferentes medios de transporte resumida en la tabla número 2, utilizando los vehículos con permiso de circulación vigente y además la flota de vehículos del transporte publico o colectivo, contabilizados y agrupados por cada comuna de una región en específico.

El hecho que esta base de datos se encuentre enfocado en las comunas, acota nuestra área de estudio, ya que no se poseen los datos granulado por calles o cuadras como en los anteriores dos datasets, generando así la necesidad de utilizar como frontera de estudio los datos comunales presentes en el portal, demarcando así nuestra área de estudio, siendo estas las comunas de Viña del Mar y Valparaíso. A continuación presentamos los datos cuantitativos a utilizar y las clases utilizadas por el INE.

Tipo de Transporte	Viña del Mar	Valparaíso
Automóvil	62806	52895
Furgón	2568	2205
Minibús	174	227
Camioneta	8677	5891
Motocicleta	4116	2140
Otros motorizados	23	12
Motocicleta sin motor	304	197
Transporte Colectivo		
Taxi simple	85	82
Taxi colectivo	1096	2163
Taxi turismo	144	58
Minibús transporte	443	305
Bus transporte	615	827

Tabla 2: Datos de vehículos para Viña del Mar y Valparaíso

Los datos demográficos anteriores son generados el año 2017 mientras que los datos del parque vehicular el 2022. La solución presente es multiplicar según el factor de crecimiento presente al año 2022 en comparación a 2017, donde se observa la población aumento en un $7,8\,\%$ en Valpararaíso y en un $10,5\,\%$ en Viña del Mar con respecto a 2017, por lo que mantenemos la distribución entregada por el Censo del 2017 pero utilizamos las magnitudes actualizadas a 2022, generando finalmente $358,839$ personas en Viña del Mar y $325,099$ en Valparaíso.

Dentro de la clasificación y generación de clases de los vehículos propuesta por la INE no se estipula una rubrica definida de manera publica, mas allá de ciertas aclaraciones para ciertas clases presentes como comentario dentro del *excell*, tales como:

- Automóvil: Incluye carrozas fúnebres y vehículos todo terreno tipo *jeep*.
- Furgón: Incluye carrozas fúnebres y ambulancias.
- Otros motorizados: Incluye casa rodante automotriz y otros.
- Minibús transporte: Incluye minibús privado remunerado, turismo, escolar, trabajadores y otros.
- Bus transporte: Incluye taxibús, bus escolar, bus de trabajadores y otros.

Debido a que se tienen los datos de diferentes clases de vehículos se asigna a cada una de ellas un costo de capacidad, representando el espacio a utilizar dentro de una calle o arco en nuestra red, y por otro lado una capacidad de asientos dirigida a las personas a contener, para esto no basta con tener en consideración las dimensiones de los vehículos sino que también

el espacio entre ellos para asignar realmente la capacidad del arco que abarcaran, en base a la representación elegida se utilizaron los siguientes valores, considerando información de las dimensiones de vehículos mas comunes e información entregada por fichas técnicas de vehículos públicos:

Tipo de Vehículo	Personas	Longitud [m]	Presencia Viña	Presencia Valparaíso
Automóvil	4	4,5	0,7749	0,7895
Furgón	10	6,5	0,0317	0,0329
Minibús	15	8,0	0,0021	0,0034
Camioneta	3	6,0	0,1071	0,0879
Motocicleta	2	2,0	0,0508	0,0319
Otros motorizados	1	3,5	0,0003	0,0002
Motocicleta sin motor	1	1,5	0,0038	0,0029
Taxi simple	4	4,5	0,0010	0,0012
Taxi colectivo	4	4,5	0,0135	0,0323
Taxi turismo	4	4,5	0,0018	0,0009
Minibús transporte	25	8,0	0,0055	0,0046
Bus transporte	40	12,0	0,0076	0,0123

Tabla 3: Capacidad de asientos, longitud y valores adicionales de vehículos

Considerando los valores de esta tabla no es negativo mencionar que se pueden comparar las longitudes y las capacidad de asientos asociados a personas, para obtener cual vehículo vendría siendo mas eficiente según la representación utilizada, ya que por ejemplo el Bus de transporte posee la mayor densidad de población frente a una camioneta, generando así que la identidad de una comuna y su parque vehicular pueda ser de gran diferencia a la hora de una evacuación y no solamente la red de transporte. Por otro lado no se considerara las diferentes velocidades que podrían llegar los diferentes vehículos sino que todos poseen el mismo costo para atravesar un arco, únicamente se diferenciaran en su costo de capacidad al usar una calle y la capacidad de personas que abarcaran de forma estática.

4.3.1. Asignación de Vehículos a nuestra red:

Dado esto utilizando la población de cada nodo se distribuyen los vehículos según el porcentaje de población que exista en ese nodo y también del porcentaje de los vehículos existentes, salvo los Minibús de transporte los cuales poseen una heurística especializada, como mencionaremos posteriormente.

Algoritmo 3 Asignar vehículos a nodos

Require: $nodes$: Lista de Nodos
Require: G: Red de Transporte
Require: car_people_dict: Diccionario asignando tipo de vehículo y su cantidad
Require: $people\_total$: Total de personas
Require: $percentage\_cars\_dict$: Diccionario asignando tipos de vehículos y su presencia porcentual
Require: $vehicle\_seating\_capacity$: Diccionario asignando tipos de vehículos a capacidad de asientos

$idnode\_to\_vehicles\_vina \leftarrow$ Diccionario Vacío
for $node\_id$ **in** $nodes$ **do**
  $personas\_in\_node \leftarrow$ Population of $node\_id$ from G
  $parque\_for\_id \leftarrow$ Empty list
  $micros\_for\_id \leftarrow 0$
  $percentage\_people\_in\_node \leftarrow$ Población de $node\_id$ divided by $people\_total$
  for $type\_of\_Car, cantidad$ **in** car_people_dict **do**
    $type\_car \leftarrow$ Extraer tipo de vehículo $type\_of\_Car$
    $seating\_capacity \leftarrow$ Capacidad de asientos $type\_car$ from $vehicle\_seating\_capacity$
    $percentage\_car\_in\_node \leftarrow$ Porcentaje de $type\_of\_Car$ in $node\_id$ from $percentage\_cars\_dict$
    if $type\_of\_Car$ is the last key in car_people_dict **then**
      if $node\_id$ is not in $paradero\_to\_node\_id$ keys **then**
        continue
      end if
      $paraderos \leftarrow$ Paraderos associated with $node\_id$
      $percentage\_stop\_paradero \leftarrow 0$
      for $paradero$ **in** $paraderos$ **do**
        $percentage\_stop\_paradero \mathrel{+}=$ Percentage of stop in $paradero$
      end for
      $micros\_for\_id \leftarrow$ Round the product of $percentage\_stop\_paradero$ and $cantidad$
    else
      $cantidad\_vehiculos \leftarrow$ Round the product of $personas\_in\_node$, $percentage\_car\_in\_node$, and $seating\_capacity$
      $cantidad\_vehiculos2 \leftarrow$ Round the product of $percentage\_people\_in\_node$ and $cantidad$
      $parque\_for\_id$.append(min($cantidad\_vehiculos$, $cantidad\_vehiculos2$))
    end if
  end for
  $parque\_for\_id$.append($micros\_for\_id$)
  $idnode\_to\_vehicles\_vina[node\_id] \leftarrow parque\_for\_id$
end for
return $idnode\_to\_vehicles\_vina$

En simples palabras se itera a través de cada nodo de la red de transporte, de el se extrae cual es el porcentaje de población presente en comparación al total de población, luego iteramos por cada tipo de auto presente en nuestras consideración, extrayendo también, la presencia porcentual como la columna de la tabla 3. Una vez obtenidos estos porcentaje para cada nodo y cada tipo de vehículo, se le asocia al nodo el mínimo valor entre estas dos ecuaciones:

- $cantidad\_vehiculos = \text{round}\left(\frac{\text{personas\_in\_node} \times \text{percentage\_car\_in\_node}}{\text{seating\_capacity}}\right)$
- $cantidad\_vehiculos2 = \text{round}\left(\text{percentage\_people\_in\_node} \times \text{cantidad}\right)$

Este método se utiliza para las dos comunas de estudio en ocasiones separadas utilizando los porcentajes de cada vehículo y totalidad de personas respectivamente.

Por otro lado se utiliza un método específico para la clase **Minibús de Transporte**, asociada a las micros de la región, debido a que su posición y distribución es factible de representar utilizando las rutas de ellas, con los datos presentes en el portal de *datasets* del gobierno, utilizando estos datos se obtiene cuales son los paraderos con mayor frecuencia dentro de los conjuntos de rutas, dado esto se itera por todas las rutas de la región de Valparaíso, contabilizando las veces que estas rutas pasan por un paradero específico, asignando finalmente un porcentaje según cuantas veces fue visitado el paradero en comparación a todas las visitas contabilizadas por todas las rutas de cada número o identificador de cada micro, luego cada paradero es asociado al arco mas cercano, se utiliza el arco debido a que los paraderos no suelen estar en intersección de calles, sino asociados a calles respectivamente, calculado la distancia punto(paradero)-linea(calle), y finalmente se asocia el paradero al nodo de origen de aquel arco. Una vez generada la asociación, se incorpora al algoritmo anterior cuando este itera sobre un nodo que sea asociado a unos de estos paraderos. Para finalmente asignar un porcentaje de Minibús de transporte según el porcentaje de presencia y la cantidad de vehículos disponibles. La estructura del *.txt* conteniendo información sobre cada paradero y sus coordenadas.

Tabla 4: Datos de Paraderos

stop_id	stop_name	stop_lat	stop_lon	location_type
89321	Falebella - Buenos Aires	-33.04771791	-71.61788796	0
89320	El Ocaso - Guaranies	-33.04915228	-71.41506923	0
89323	Ricardo Cumming 883	-33.05698703	-71.44125913	0
89322	Santiago 575 - 593	-33.03806553	-71.4052395	0

Profundizando en los detalles de **Minibús de Transporte** utilizando el dataset y obteniendo los paraderos asociados a las 2 comunas de estudio realizamos el siguiente algoritmo para asignar los paraderos a nodos en base a los arcos.

Algoritmo 4 Asignación de Paraderos a Nodos

```
paradero_to_node_id ← Diccionario vacío
stop_id_interest ← Conjunto vacío
for cada fila row en stops do
    lat ← row['stop_lat']
    lon ← row['stop_lon']
    Crear un punto paradero con coordenadas (lon, lat)
    if polygono_vina2 contiene paradero o polygono_valpo2 contiene paradero then
        Agregar row['stop_id'] a stop_id_interest
        id_node_close_to_paradero ← Nodo más cercano a paradero en G
        if id_node_close_to_paradero está en las claves de paradero_to_node_id then
            Agregar row['stop_id'] a paradero_to_node_id[id_node_close_to_paradero]
        else
            paradero_to_node_id[id_node_close_to_paradero] ← [row['stop_id']]
        end if
    end if
end for
for cada fila row en stops do
    lat ← row['stop_lat']
    lon ← row['stop_lon']
    Crear un punto paradero con coordenadas (lon, lat)
    if polygono_vina2 contiene paradero o polygono_valpo2 contiene paradero then
        Agregar row['stop_id'] a stop_id_interest
    end if
end for
```

Para luego obtener los porcentajes anteriormente mencionados contando las veces que cada recorrido pasa por los paraderos obtenidos, utilizando únicamente un recorrido para cada Minibús de transporte. Además cada para o *stop* como es mencionada en el mismo nombre del *.txt* que es como dataset posee la siguiente estructura, entre otras columnas que no son de interés para el caso de estudio.

Tabla 5: Datos de stops_times

trip_id	arrival_time	departureTime	stop_id	stopSequence	pickup_type
244404S9802P1	06:00:00	06:00:00	86920	1	0
244404S9802P1	06:00:20	06:00:20	86921	2	0
244404S9802P1	06:00:50	06:00:50	86922	3	0
244404S9802P1	06:01:20	06:01:20	86923	4	0

Algoritmo 5 Asignación de porcentajes a cada nodo

```
count_for_stops ← Diccionario vacío
percentage_stop ← Diccionario vacío
for cada fila row en stop_time do
    trip_id ← row['trip_id']
    vuelta ← Último segmento de trip_id separado por 'P'
    if int(vuelta) ≠ 1 then
        Continuar con la siguiente iteración del bucle
    end if
    stop_id ← row['stop_id']
    if stop_id no está en stop_id_interest then
        Continuar con la siguiente iteración del bucle
    end if
    if stop_id está en las claves de count_for_stops then
        count_for_stops[stop_id] += 1
    else
        count_for_stops[stop_id] ← 1
    end if
end for
total_stop_usage ← suma de los valores en count_for_stops
types_of_values ← Conjunto vacío
for cada par stop, count_stop en count_for_stops do
    Agregar count_stop a types_of_values
    percentage_stop[stop] ← count_stop/total_stop_usage
end for
```

Utilizando este método se obtiene que no todos los vehículos serán usados, al igual que no todas las personas se encontraran dentro de un vehículo, lo cual es algo considerado natural dentro de la naturaleza de una evacuación, es más, no siempre sera la mejor opción utilizar un transporte para evacuar, pero esto va mas allá del alcance de la memoria que se enfoca principalmente en la preparación y búsqueda de debilidades de nuestra red de transporte. Finalmente se tiene que en Viña del Mar de 81051 vehículos, solamente se utilizaran para la evacuación 75236, y en Valparaíso 62116 de 67002.

Finalmente se obtendremos asociados una lista para cada nodo, donde cada valor de ella corresponde la cantidad de vehículos de una clase especifica existentes en esa localidad.

De igual manera que se genero una metodología para los Minibús de transporte, sería positivo utilizar datos que entreguen información sobre la distribución de vehículos en particular, como por ejemplo incorporar una nueva columna en el censo que nos presente la cantidad de vehículos en la cuadra, u obtener una mejor descripción en detalle de las clases entregadas por el parque vehicular.

4.4. Representación de solución:

Implementadas las técnicas de estimación de distribución demográficas y asociada la población a diferentes vehículos utilizando el parque vehicular de la región, se puede definir el plan de evacuación para cada uno de los vehículos obtenidos dentro de su posición asignada.

Esta sección expresa la metodología usada y las característica de cada paso al construir el método iterativo para optimizar las rutas de cada vehículo con el fin de minimizar el tiempo de evacuación final. Esto bajo la premisa que al obtener el mínimo tiempo a realizar da a conocer las debilidades de la red de transporte de manera específica, siendo una prueba de estrés eficaz, ya que estudia cuales son los arcos o conjunto de vehículos mas afectados para el caso particular, al igual de cuales fueron los puntos de evacuación mas utilizados o de mayor preferencia para el algoritmo y su solución final.

La forma en que se representa la solución es un diccionario donde cada vehículo sera asociado a una llave única y contendrá una lista con los nodos por lo cual transitara hasta evacuar de la zona de peligro. Dado que el enfoque es únicamente en los vehículos en peligro no es necesario tener en cuenta los 137352 vehículos presentes en estas dos comunas, sino que solamente existen 19742 vehículos en peligro.

Para inicializar la representación de solución se utiliza el método *greedy* de asignar a cada vehículo la ruta con menor costo de tiempo, un enfoque optimista, esta variable la llamaremos *routes_cars*.

4.4.1. Creación de Rutas

1. ***Inicialización de Variables***:
    - Se inicializan variables para almacenar las rutas de los autos (`routes_cars`), el número total de movimientos a realizar (`movimientos_a_realizar`), y la capacidad total de asientos de los autos (`seating_capacity_car`), con el fin de observar la cantidad de personas que se trasladaran.
2. ***Iteración sobre los Nodos y Autos por Nodo***:
    - Para cada nodo (`node_id`) y la cantidad de autos en ese nodo (`cars_count_by_node`), estructura descrita en el paso anterior, se verifica si el nodo está en peligro utilizando (`nodes_a_salvo_set`) y no es igual a *node_dummy_danger*. En caso contrario, se continúa con el siguiente nodo, con el fin de obtener únicamente los vehículos necesarios a evacuar y así simplificar el problema, el cual de por si requiere de un costo de computo y memoria elevado.
3. ***Generación de Rutas***:

- Se obtiene la ruta desde el nodo actual hasta el punto de evacuación (`ruta`) desde la matriz de costos de ruta (`ruta_mínima`), esta matriz de costo fue calculada utilizando *Dijkstra*, el cual es el método por defecto de la librería *networkx*, la cual abarca una gran variedad de algoritmos enfocado a grafos.
- Se itera sobre los diferentes tipos de vehículos y la cantidad de cada tipo, ya que como se menciona anteriormente se tiene una lista con la cantidad de vehículos de cada tipo.
- Se genera una clave única para cada auto, que incluye un contador, el ID del nodo, y el tipo de vehículo, cabe destacar que los últimos dos valores se utilizan únicamente como *foreign_key*, con el fin de trasladar la información de manera eficiente sin requerir de estructuras especiales, al momento de la evacuación.
- Se acumula la capacidad de asientos del tipo de vehículo, equivalente a las personas que se evacuaran, según el tipo de vehículo y representada en la tabla 3.

4. ***Actualización de Variables de Movimiento***:
    - Se actualiza el contador de ID y el número total de movimientos a realizar.
5. ***Salida***:
    - La función entrega las siguientes tres estructuras:
        - `routes_cars`: Un diccionario que asigna una clave única a cada auto y su ruta correspondiente.
        - `movimientos_a_realizar`: El número total de movimientos que se realizarán en todas las rutas de los autos.
        - `seating_capacity_car`: La capacidad total de asientos de todos los autos generados.

A continuación se presenta el pseudocódigo utilizado para la generación de rutas y descrito anteriormente con mayor detalle, debido a que sera la base para los siguientes algoritmos, ya que se trabaja sobre aquella estructura, ya que es la representación de solución, es decir la ruta para cada vehículo.

Algoritmo 6 Generar Rutas de Autos Menor Costo

Require: $cars\_count\_by\_node$: Diccionario de conteo de autos por nodo,realizado en la asignación de vehículos del algoritmo 3
Require: $nodes\_a\_salvo\_set$: Conjunto de nodos a salvo
Require: min_path_costs: Rutas de menor costo para cada nodo
Require: $vehicle\_seating\_capacity$: Diccionario de capacidad de asientos por tipo de vehículo

$routes\_cars \leftarrow$
$movimientos\_a\_realizar \leftarrow 0$
$contador\_id \leftarrow 0$
$seating\_capacity\_car \leftarrow 0$
for $node\_id, cars$ **en** $cars\_count\_by\_node$ **do**
  if $node\_id$ no está en $nodes\_a\_salvo\_set$ **o** $node\_id \ ==' \ node\_dummy\_danger'$ **then**
    continuar
  end if
  $ruta \leftarrow ruta\_minima[node\_id]$
  for $type, vehicle\_type$ **en** enumerar($cars$) **do**
    for $cantidad\_of\_car\_type$ **en** rango($vehicle\_type$) **do**
      $key\_car \leftarrow f"contador\_id\_node\_id\_list(vehicle\_seating\_capacity.keys())[type]$

      $seating\_capacity\_car+ = list(vehicle\_seating\_capacity.values())[type]$
      $routes\_cars[key\_Car] \leftarrow ruta$
      $contador\_id+ = 1$
      $movimientos\_a\_realizar+ =$ longitud de $ruta - 2$
    end for
  end for
end for
return $routes\_cars, movimientos\_a\_realizar, seating\_capacity\_car$

4.5. Simulación Evacuación:

4.5.1. Organización de las rutas

El siguiente paso se utiliza para organizar de una manera eficiente los movimientos a realizar de cada ruta de los vehículos, con el fin de realizar los movimientos en orden y saber directamente cuales son los vehículos que tendrán movimientos en un destinado tiempo. La eficiencia de por si es algo deseable en toda implementación, pero esto es aún mas deseable al utilizar métodos iterativos, ya que esta simulación se llevara a cabo en múltiples instancias, siendo de vital importancia una implementación eficiente, con el fin de poder generar una mayor cantidad de iteraciones en un menor tiempo.

Para poder realizar este objetivo se implementaron estructuras que almacenan las marcas de tiempo para los movimientos de los autos a lo largo de la evacuación. generando así, la siguiente transformación a partir de las variables y representación de solución inicial *routes_cars*. El cual toma como entrada el grafo que representa la red de transporte que llamaremos(**G**) y las rutas para cada auto (`routes_cars`) como mencionamos en el paso anterior.

1. **Inicialización**: Se inicializan diccionarios para almacenar marcas de tiempo de los movimientos para cada auto, (`car_timestamp_mov`), todas las marcas de tiempo de los movimientos (`all_timestamp_of_movement`), y los autos presentes en cada marca de tiempo (`car_in_timestamp`).
    - ***car_timestamp_mov:*** Diccionario donde cada vehículos posee su llave, en la cual se almacena otro diccionario, que posee como llave los tiempos en lo que se genera cierto movimiento, cada movimiento se representa como [id_origen,id_destino,id_nodo_subsiguiente,tipo_de_vehiculo].
    - ***all_timestamp_of_movement:*** Conjunto que almacena todos los tiempos donde se genera algún movimiento en la red.
    - ***car_in_timestamp:***Diccionario que posee todos los autos que presentan un movimiento dado cierto tiempo.
2. **Iteración Sobre las Rutas de los Autos**:
    - Para cada vehículo y su ruta correspondiente (`ruta`), la función procesa los movimientos.
    - Descompone el ID del auto para identificar su tipo.
    - Inicializa variables para rastrear el costo total y la longitud de la ruta.
3. **Procesamiento de la Ruta**:
    - Itera sobre cada segmento de la ruta, es decir cada nodo presente en la ruta, calculando el costo total de cada calle, según su distancia, mencionado dentro de las características de cada arco.
    - Construye un diccionario de marcas de tiempo para cada movimiento, registrando el costo del movimiento, el arco actual (`id_origen`), el siguiente arco (`id_destino`), y el arco subsiguiente (`id_nodo_subsiguiente`), junto con el tipo de auto, como *foreign key*, ya que así podemos obtener en la simulación el movimiento de personas y el costo de capacidad respectivo. Si bien, se almacenan 3 arcos para un movimiento que únicamente concierne a los primeros, esto es con el fin de movilizar el vehículo y por ende el costo de capacidad al segundo par una vez finalizado el movimiento actualizar la red correctamente.
    - Se actualiza el diccionario en cada *timestamp* de la ruta con el id del vehículo en `car_in_timestamp` para rastrear qué autos están presentes en cada marca de tiempo, útil al iterar cronológicamente en la evacuación sin la necesidad de iterar sobre todos los vehículos.

- Agregamos cada costo de movimiento al conjunto de todas las marcas de tiempo, con el fin de iterar solamente en tiempos útiles. Cabe destacar que al inicio de la simulación existen movimientos en gran cantidad de los tiempos pero a medida que disminuyen los vehículos es de mayor provecho ya que existen mayor diferencia entro los movimientos y sus tiempos, mejorando la eficacia al seleccionar los movimientos y evitar tener que iterar por cada valor de cada textittimestamp.

4. **Finalización y Ordenamiento**:
   - Se registran movimientos para un auto, agrega el diccionario de marcas de tiempo a `car_timestamp_mov`.
   - Ordena y convierte el conjunto de todas las marcas de tiempo en una lista ordenada, para posterior recorrido en orden cronológico.
5. **Salida**:
   - La función devuelve tres salidas principales:
     - `car_timestamp_mov`: Un diccionario que asigna cada auto a sus movimientos de marca de tiempo correspondiente.
     - `all_timestamp_of_movement`: Una lista ordenada que contiene todas las marcas de tiempo que presenten movimientos, en una primera instancia, ya que posteriormente debido a los conflictos de capacidad se agregan mayor cantidad de textittimestamp.
     - `car_in_timestamp`: Un diccionario que asigna cada marca de tiempo a la lista de autos presentes en ese textittimestamp, el cual igual sera modificado a lo largo de la evacuación debido a que los vehículos se retrasan por conflictos de capacidad.

A continuación se da a conocer el pseudocódigo asociado a la función utilizada para la implementación de las estructuras anteriores, donde se pueden observar con mayor detalles las sutilezas que esta presenta además de la fidelidad del mismo código utilizado.

Algoritmo 7 Obtener Marca de Tiempo de Evacuación

Require: G: Grafo de la red de carreteras
Require: $routes\_cars$: Diccionario de rutas de autos
Require: $nodes\_inside\_polygons\_set$: Conjunto de nodos dentro de polígonos

$car\_timestamp\_mov \leftarrow$ Diccionario Vacío
$all\_timestamp\_of\_movement \leftarrow$ Conjunto Vacío
$car\_in\_timestamp \leftarrow$ Diccionario Vacío
for auto, ruta **en** $routes\_cars$ **do**
  $timestamp\_mov \leftarrow$ Diccionario Vacío
  $type\_car \leftarrow$ Unir los últimos elementos de $auto$ separados por '_'
  $iterador \leftarrow 0$
  $cost\_total \leftarrow 0$
  $largo\_ruta \leftarrow$ Longitud de $ruta$
  while $iterador < largo\_ruta - 2$ **do**
    $id\_from, id\_to, id\_next\_to \leftarrow ruta[iterador], ruta[iterador + 1], ruta[iterador + 2]$
    $cost\_total+ =$ Entero de $G.edges[id\_from, id\_to]['distancia']$
    $timestamp\_mov[cost\_total] \leftarrow [id\_from, id\_to, id\_next\_to, type\_car]$
    if $cost\_total$ está en las claves de $car\_in\_timestamp$ **then**
      $car\_in\_timestamp[cost\_total] \mathrel{+}= [auto]$
    else
      $car\_in\_timestamp[cost\_total] \leftarrow [auto]$
    end if
    Agregar $cost\_total$ a $all\_timestamp\_of\_movement$
    $iterador \mathrel{+}= 1$
  end while
  if Existe $timestamp\_mov$ **then**
    $car\_timestamp\_mov[auto] \leftarrow timestamp\_mov$
  end if
end for
Ordenar $all\_timestamp\_of\_movement$
return $car\_timestamp\_mov$, $all\_timestamp\_of\_movement$, $car\_in\_timestamp$

4.6. Realización de movimientos

Ahora al obtener las 3 estructuras se lleva a cabo los movimientos de cada vehículo de manera eficiente minimizando la búsqueda de vehículos que no poseen movimientos al igual que tiempos donde existen aquellos, llevando así a cabo la evacuación por orden de tiempo cronológico, pero ahora se da a conocer la naturaleza y complejidad, al igual que el punto de mayor énfasis e importancia dentro de esta, el surgimiento de atascamiento en ciertas calles. Aquello naturalmente lleva a que ciertos vehículos se vean retrasados y los tiempos de sus rutas se vean por ende modificados en sus movimientos posteriores frente al estanca-

miento específico. Dado esto la metodología usada al transcurrir el movimiento del vehículo al cruzar una calle y desencadenado un conflicto de capacidad, desencadenando una penalización de tiempo igual a la cantidad de exceso de capacidad que posea la calle, esto quiere decir que si por ejemplo al intentar incluir cierto vehículo en una calle y este es el primer en generar un conflicto de capacidad su penalización sera menor que si la calle ya se encuentra con el doble de su capacidad excedida, generando así para el segundo caso una penalización de mayor magnitud, ya que que la calle se encuentra en el transcurso de un mayor atascamiento. Cabe destacar que la penalización utilizada es lineal, por lo que la penalización es igual a la sobrecarga de la calle, únicamente teniendo como consideración la sobrecarga de ella generada por las capacidades de los vehículos correspondientes.

Todos los pasos anteriores se encuentran definidos en el algoritmo *simulate_evacuation*, encargado de simular la evacuación de vehículos directamente en la red de transporte, movilizando principalmente por sencillez la ocupación de cada arco, ya que por ejemplo movilizar la población o el atributo de cada arco con la cantidad de ese vehículo en cada clases es innecesario. A continuación, se detalla cada parte del algoritmo, utilizando las estructuras anteriormente mencionadas:

4.6.1. Inicialización de Variables

- $conflict$: Variable que almacena el número total de conflictos encontrados durante la evacuación, utilizada únicamente como KPI (*Key Process Information*), y variable dentro de nuestra función objetivo.
- $expedito$: Variable que cuenta el número total de movimientos realizados hasta el momento.
- $conflictos\_by\_car$: Diccionario que registra el número de conflictos por cada vehículo.
- $tiempo\_finishes\_car$: Diccionario que mapea los vehículos a la marca de tiempo en la que completaron su evacuación, con el fin de obtener el tiempo final de evacuación de cada vehículo.
- G: Grafo utilizado para la red de transporte.
- all_cars: Lista que contiene todos los identificadores de vehículos.
- $left\_cars$: Variable que registra la cantidad restante de vehículos que no han completado su evacuación, la cual sera la condición de termino y también sera constantemente modificada al eliminar los vehículos finalizados.

Cabe destacar que existen valores que utilizaremos luego para el análisis de resultados y exploratorio de una evacuación final.

4.6.2. Bucle Principal

El bucle principal continúa hasta que todos los vehículos hayan completado su evacuación o no queden más marcas de tiempo de movimiento.

4.6.3. Iteración por Marcas de Tiempo

- Se obtiene la marca de tiempo actual y se recorren todos los vehículos que tienen movimientos en esa marca de tiempo.
- Para cada vehículo, se verifica si hay conflicto en la siguiente calle de su ruta.
- Si hay conflicto, se actualizan las marcas de tiempo de movimiento de los vehículos afectados.
- Si no hay conflicto, se mueve el vehículo a la siguiente calle y se actualiza la ocupación de las calles involucradas.

4.6.4. Finalización de la Evacuación

Una vez que todos los vehículos han completado su evacuación, se obtiene la información relevante, incluyendo la marca de tiempo final, el número total de conflictos, los conflictos por vehículo, los vehículos finalizados y las marcas de tiempo de finalización de cada uno de estos vehículos.

4.6.5. Detalles de Evacuación

El algoritmo simula eficazmente la evacuación de vehículos en una red de carreteras durante una emergencia. Utiliza un enfoque basado en marcas de tiempo para coordinar los movimientos de los vehículos y gestionar los conflictos entre ellos. La copia profunda del grafo original asegura que los cambios en la red de carreteras durante la evacuación no afecten la simulación. Además, el algoritmo maneja adecuadamente los casos de vehículos que completan su evacuación y los errores o conflicto de capacidad que puedan surgir durante la simulación.

Algoritmo 8 Simular Evacuación

Require: $car\_timestamp\_mov$: Diccionario de marca de tiempo de movimiento de autos
Require: $vehicle\_seating\_capacity$: Diccionario de capacidad de asientos por tipo de vehículo
Require: $nodes\_inside\_polygons\_set$: Conjunto de aristas dentro de polígonos
Require: $all\_timestamp\_of\_movement$: Lista de todas las marcas de tiempo de movimiento
Require: $car\_in\_timestamp$: Diccionario de autos en marcas de tiempo

$conflict \leftarrow 0$
$expedito \leftarrow 0$
$conflictos\_by\_car \leftarrow$ Diccionario Vacío
$finished\_cars \leftarrow$ Lista Vacía
$tiempo\_finishes\_car \leftarrow$ Diccionario Vacío
$G \leftarrow$ Copia profunda de $G\_original$
$all\_cars \leftarrow$ Lista de todas los autos
$left\_cars \leftarrow$ Longitud de all_cars
while $left\_cars > 0$ **y** $len(all\_timestamp\_of\_movement) > 0$ **do**
  $time\_actual \leftarrow$ Marca de tiempo actual de $all\_timestamp\_of\_movement$.pop(0)
  for auto **en** $car\_in\_timestamp[time\_actual]$ **do**
    $timestamp\_mov \leftarrow car\_timestamp\_mov[auto]$
    $id\_from, id\_to, id\_next\_to, car\_type \leftarrow timestamp\_mov[time\_actual]$
    $capacity\_cost \leftarrow vehicle\_lengths[car\_type]$
    $taco \leftarrow$ Máximo(0, $G.edges[id\_to, id\_next\_to]['ocupacion'] + capacity\_cost - G.edges[id\_to, id\_next\_to]['capacity']$)
    if $taco > 0$ **then**
      $conflict \mathrel{+}= taco$
      $car\_timestamp\_mov, all\_timestamp\_of\_movement, car\_in\_timestamp \leftarrow$ **update_time_routes**($car\_timestamp\_mov$, $time\_actual$, $taco$, $all\_timestamp\_of\_movement$, $car\_in\_timestamp$, $auto$)
    else
      $expedito \mathrel{+}= 1$
      $G.edges[id\_from, id\_to]['ocupacion'] \mathrel{-}= capacity\_cost$
      if $G.edges[id\_from, id\_to]['ocupacion'] < 0$ **then**
        $G.edges[id\_from, id\_to]['ocupacion'] \leftarrow 0$
      end if
      $G.edges[id\_to, id\_next\_to]['ocupacion'] \mathrel{+}= capacity\_cost$
      if $id\_next\_to ==' nodes\_outside\_polygons'$ **then**
        all_cars.remove($auto$)
        Eliminar $auto$ de $car\_timestamp\_mov$
        $tiempo\_finishes\_car[auto] \leftarrow time\_actual$
        continuar
      end if
      Eliminar $time\_actual$ de $car\_timestamp\_mov[auto]$
    end if
  end for
  $left\_cars \leftarrow$ Longitud de all_cars
end while
return $time\_actual, conflict, conflictos\_by\_car, finished\_cars, tiempo\_finishes\_car$

Como se observa cada vez que se intenta ejecutar el movimiento, o en este caso modificar la ocupación de un arco este puede aumentar la capacidad propuesta por lo que se llama la función *update_time_routes*, la cual tiene como principal objetivo modificar la ruta del vehículo perteneciente, además de actualizar las estructuras de ayuda mencionadas anteriormente, modificando los tiempos de ese vehículo y agregando en caso de ser necesario nuevos tiempos donde existan movimientos.

Por otro lado debido a que todas las rutas terminan en el nodo *dummy*, *node_dummy_safe*, cuando se aproxima al nodo a salvo se genera inmediatamente el movimiento final y se elimina de las estructuras necesarias para ahorrar memoria y tiempo de computo, además de extraer los KPI, pertenecientes, almacenando el tiempo de evacuación respectivo de ese vehículo.

Algoritmo 9 Actualizar Tiempo en Rutas

Require: $car\_timestamp\_mov$: Diccionario de marca de tiempo de movimiento de autos
Require: $tiempo\_actual$: tiempo_actual, con el fin de modificar únicamente la ruta posterior
Require: $jump$: Salto en el tiempo
Require: $all\_timestamp\_of\_movement$: Lista ordenada de marcas de tiempo
Require: $car\_in\_timestamp$: Diccionario de autos en marcas de tiempo
Require: $auto$: Identificador del auto

$dictionary \leftarrow car\_timestamp\_mov[auto]$
$keys\_to\_update \leftarrow$ Lista de claves en $dictionary$ excepto $tiempo\_actual$
for $tiempo$ **en** $keys\_to\_update$ **do**
    $new\_timestamp \leftarrow tiempo + jump$
    Remover $auto$ de $car\_in\_timestamp[tiempo]$
    if $new\_timestamp$ está en las claves de $car\_in\_timestamp$ **then**
        Agregar $auto$ a $car\_in\_timestamp[new\_timestamp]$
    else
        Crear una nueva lista con $auto$ y asignarla a $car\_in\_timestamp[new\_timestamp]$
    end if
    Asignar $dictionary[tiempo]$ a $car\_timestamp\_mov[auto][new\_timestamp]$
    Remover $tiempo$ de $dictionary$
    $all\_timestamp\_of\_movement \leftarrow$ Añadir a la lista ordenada $new\_timestamp$
end for
return $car\_timestamp\_mov$, $all\_timestamp_of\_movement$, $car\_in\_timestamp$

La función anterior será la función con mayor llamadas dentro de nuestro algoritmo de evacuación siendo llamada cada vez que existen conflictos de capacidad.

4.7. Flujo Algoritmo Iterativo

Finalmente una vez realizado el algoritmo de evacuación y la representación de nuestra solución, nos enfocamos a generar el algoritmo iterativo en este caso *Hill Climbing* con alguna mejora, para esto se define la función objetivo y como se consideran mismos tiempos de evacuación para diferente conjunto de rutas, cual sera nuestro movimiento frente a este conjunto de rutas y cuantos movimientos se utilizaran para cada iteración y prueba. A continuación se demuestra el algoritmo global de la implementación.

Algoritmo 10 Algoritmo de Generación de Rutas Óptimas *(Hill Climbing)*

$generation \leftarrow 0$
while $generation < num\_of\_generation$ **do**
  for key_car **en** $car\_keys\_to\_change$ **do**
    $car\_key\_to\_modify, modified\_path \leftarrow$ Crear variación de rutas de autos
    $routes\_cars[car\_key\_to\_modify] \leftarrow modified\_path$
  end for
  $EvacuationTime, conflict, conflictos\_by\_car, finished\_cars, tiempo\_finishes\_car \leftarrow$ Generar simulación de evacuación
  if $EvacuationTime$ en $leaderboard\_time.keys()$ **then**
    if $leaderboard\_time[EvacuationTime][1] > conflict$ **then**
      $leaderboard\_time[EvacuationTime] \leftarrow [routes\_cars, conflict, conflictos\_by\_car, finished\_$
    end if
  else
    $leaderboard\_time[EvacuationTime] \leftarrow [routes\_cars, conflict, conflictos\_by\_car, finished\_ca$
    $best\_case \leftarrow$ Mejor caso en $leaderboard\_time$
  end if
  Incrementar la generación: $generation+ = 1$
end while

Como se observa la función de objetivo se basa en gran medida en el tiempo de evacuación (*EvacuationTime*) como primer argumento, este se mide al obtener el vehículo con mayor tiempo de evacuación o último vehículo en evacuar. Luego si es que obtenemos mismo tiempo de evacuación pero se observa una menor cantidad de conflictos(*Conflict*) durante la evacuación se prioriza la instancia con menor conflicto como nueva representación de solución y por ende sujeto de cambio.

Luego se genera una cantidad de movimientos o mutación a la variable *route_cars*. El movimiento se basa en modificar una sección de la ruta, esto se hace sustituyendo una porción en base a dos movimientos y una probabilidad que elija entre alguna de ellas, pero primero se elige al azar los índices de la porción y desencadenando el cambio a esa porción, con la consideración que debe finalizar conectando al siguiente nodo de la ruta que no fue considerada dentro de la porción, generando así una ruta sin destruir la ruta lógica, o sea, aceptable

por la red de transporte y que respete las direcciones de las calles.

Por lo que se define el porcentajes que representa con que predominancia se elegirán cada movimiento, ese porcentaje esta distribuido de manera uniforme entre $(0,3y1)$. Dado esto se procede a reconstruir la ruta nodo por nodo, generando un nuevo número al azar, el cual compararemos con el valor mencionado anteriormente, observando si es mayor a aquel, si es que es mayor entonces se tomara al azar como nueva ruta un vecino aleatorio de ese nodo, considerando los nodos que sean considerados destino de nuestro nodo actual, en caso que el valor sea menor, entonces se tomara el nodo vecino destino que sea parte de la ruta con menor costo, en dirección al siguiente nodo que no es parte de nuestra porción como siguiente nodo en nuestra construcción.

Algoritmo 11 Función para Crear Variación en Rutas

Require: $routes\_cars$: Diccionario de rutas de autos
Require: G: Grafo de la red de transporte
Require: $nodes\_inside\_polygons\_set$: Conjunto de bordes fuera de polígonos
Require: key_car: Clave del auto a modificar

$random\_value \leftarrow key\_car$
$ruta \leftarrow []$
$original\_path \leftarrow$ Copiar la ruta original del auto
if len$(original_path) \leq 4$ **then**
    return $random\_value, original\_path$
end if
$replace\_index \leftarrow$ Entero aleatorio entre 1 y len$(original_path) - 4$
$target\_index \leftarrow$ Entero aleatorio entre $replace\_index + 2$ y len$(original_path) - 2$
$target \leftarrow$ Nodo objetivo en la ruta original en el índice $target\_index$
$replace\_node \leftarrow$ Nodo a reemplazar en la ruta original en el índice $replace\_index$
$ruta, normal \leftarrow$ Cambiar porción de la ruta
if normal **then**
    $modified\_path \leftarrow original\_path[: replace\_index + 1] + ruta[1 :] + original\_path[target\_index + 1 :]$
else
    $modified\_path \leftarrow original\_path[: replace\_index + 1] + ruta[1 :]$
end if
return $random\_value, modified\_path$

Inicialmente se consideraba utilizar únicamente movimiento al azar, pero este generaba rutas con gran costo y que terminaban siendo contraproducente al tener que sobrellevar una gran cantidad de iteraciones para encontrar alguna mejor, por lo que se genero un gradiente hacia el destino deseado, esto se hace incorporando el porcentaje de elegir el siguiente nodo en dirección al destino con menor costo, el cual igual posee y genera movimientos únicos y azarosos capaz de abarcar potencialmente todo el espacio de búsqueda de rutas para el *Hill Climbing*, pero con la salvedad de este gradiente como mencionamos.

Otra consideración a tener en cuenta es que debido que a se genera una ruta aleatoria y debido a la gran cantidad de rutas presentes puede ocurrir a que al modificar esta porción de ruta el mismo cambio termine generando inmediatamente una ruta la cual ya se dirige hacia un nodo que este salvo por lo que dentro la siguiente función especifica tenemos en cuenta este caso y utilizaremos esa porción como el nuevo fin de ruta.

Algoritmo 12 Función para Cambiar una Porción de la Ruta

Require: G: Grafo de la red de transporte
Require: $source$: Nodo de origen de la porción de la ruta
Require: $nodes\_outside\_polygons\_set$: Conjunto de nodos fuera de polígonos
Require: $target$: Nodo objetivo al que se desea llegar

$path \leftarrow [source]$
$current\_node \leftarrow source$
$next\_node \leftarrow$ Nodo siguiente en la ruta
$shortest\_path\_percentage \leftarrow$ Número aleatorio entre 0.2 y 1
while $current\_node \neq target$ **do**
  if $current\_node$ está en $nodes\_outisde\_polygons\_set$ **then**
    $path.append('edges\_outside\_polygons')$
    return $path$, **False**
  end if
  $neighbors \leftarrow$ Lista de arcos salientes desde $current\_node$
  if valor aleatorio de [0.1] $< shortest\_path\_percentage$ **then**
    $next\_node \leftarrow$ Nodo siguiente en el camino más corto desde $current\_node$ hasta $target$
  else
    $next\_node \leftarrow$ Nodo aleatorio entre los vecinos de $current\_node$
    while $next\_node == path[-1]$ **do**
      $next\_node \leftarrow$ Nodo aleatorio entre los vecinos de $current\_node$
    end while
  end if
  $path.append(next\_node)$
  $current\_node \leftarrow next\_node$
end while
return $path$, **True**

4.8. Análisis de resultados, mejora de rutas

Dentro de esta sección se analizara cuales fueron los resultados asociados a los costos de evacuación utilizando el algoritmo iterativo enfocado en la modificación de rutas utilizando la red de transporte original de nuestras dos comunas de estudios, al igual que la utilización de un algoritmo iterativo con el fin de modificar esta red de transporte obteniendo una red especializada para nuestro caso de estudio, generando así modificaciones versátiles para el

caso de estudio.

4.8.1. Red de Transporte Original

Lo primero a observar es cuales fueron los nodos que obtuvieron un mayor tiempo de evacuación, naturalmente, por lo que se genero un conjunto de mapas coropléticos que al representar una mayor saturación de color equivale a un mayor costo, cabe considerar además que existen múltiples vehículos para cada nodo asociado a su polígono de voronoi, por lo que se considera el promedio de costo de todos los vehículos presentes, para el sombreado del polígono.

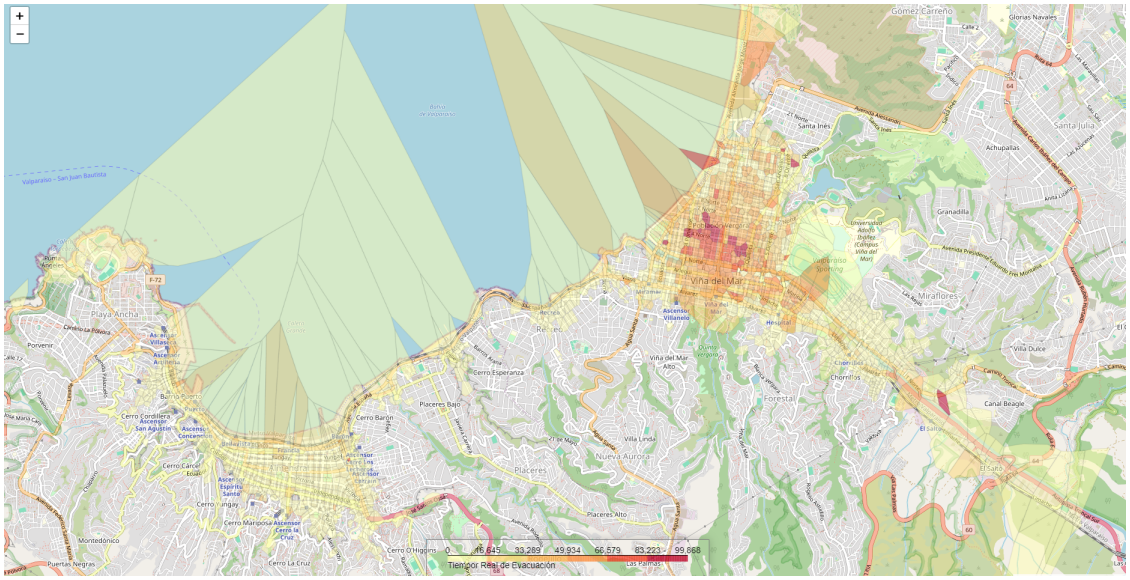


Figura 11: Costo de Evacuación Real
Fuente: Elaboración propia

Como se observa en el mapa 11, se representan los costos asociados a cada nodo en una escala de amarillo a rojo, donde a primera vista se observa que los mayores tiempos se vieron en la comuna de Viña del Mar, específicamente en el centro de viña, principalmente generando una zona roja en el centro, del plan o centro comercial, valga la redundancia, en aproximadamente el cuadrado generado por la calles desde 2 norte hasta 6 norte y desde 4 poniente hasta la calle Quillota. Observado a través de un zoom en detalle en el siguiente mapa.

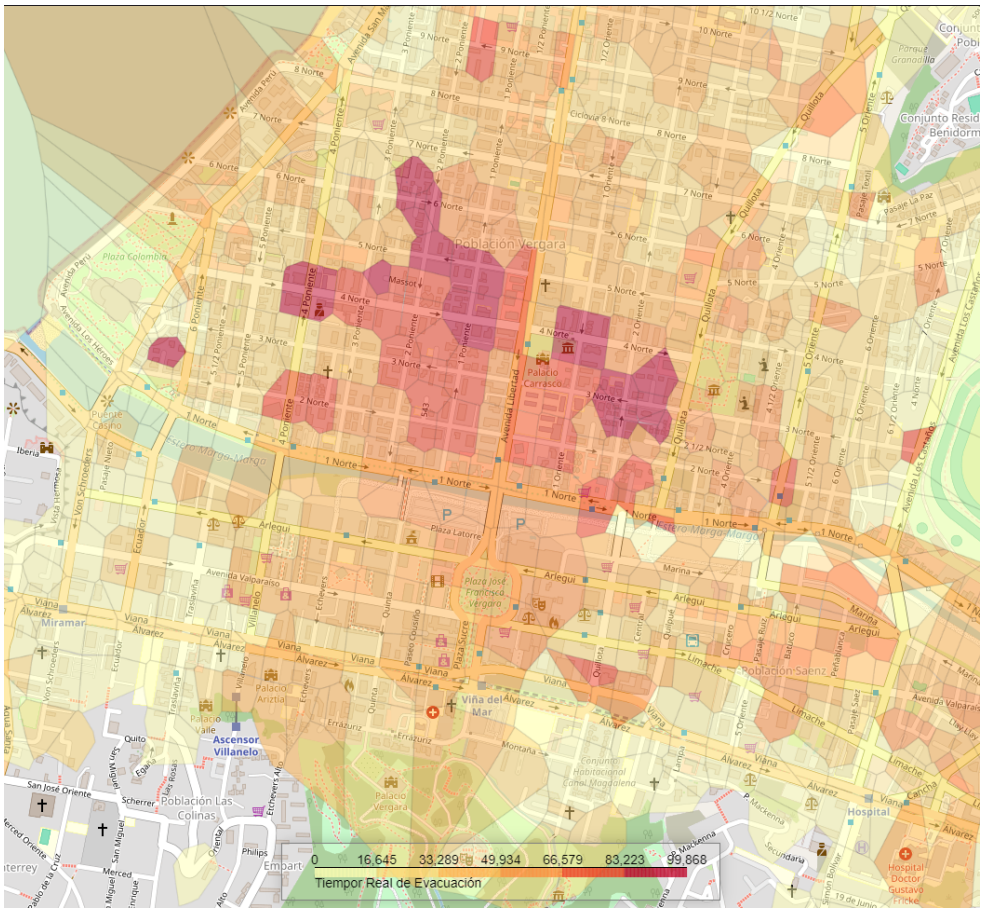


Figura 12: Costo de Evacuación Real Centro de Viña
Fuente: Elaboración propia

Los tiempos de todos los vehículos se encuentran en el grafo 13 de todos los vehículos independientemente, debemos tener en consideración y recordar a que esta asociado el costo de una ruta, ya que cada arco posee como costo su distancia donde cada unidad esta asociado a un largo de $10[cm]$, por lo que un costo de 1000 significa una calle de $100[m]$, y las penalizaciones asociadas igualmente a este costo dependen de cuantos vehículos y su tipo correspondiente sobrepasaron la calle, simulando así que el vehículo tuvo que recorrer un mayor tiempo o distancia debido a la relación que se genero para representar los atascamientos, por lo que en este caso al considerar que el vehículo con mayor costo equivalente a 90843, podría ser relacionado como un costo de tiempo de realizar $9{,}0843[km]$, mientras que nuestra solución greedy inicial nos entrega un costo de 95105.

Por otro lado también se observa que el Cerro Castillo al ser una zona segura, genera un radio sobre el que minimiza los costos de traslado a su alrededor, disminuyendo la carga de la red de transporte en otros sectores, siendo de gran relevancia para la ciudad y de gran ayuda.

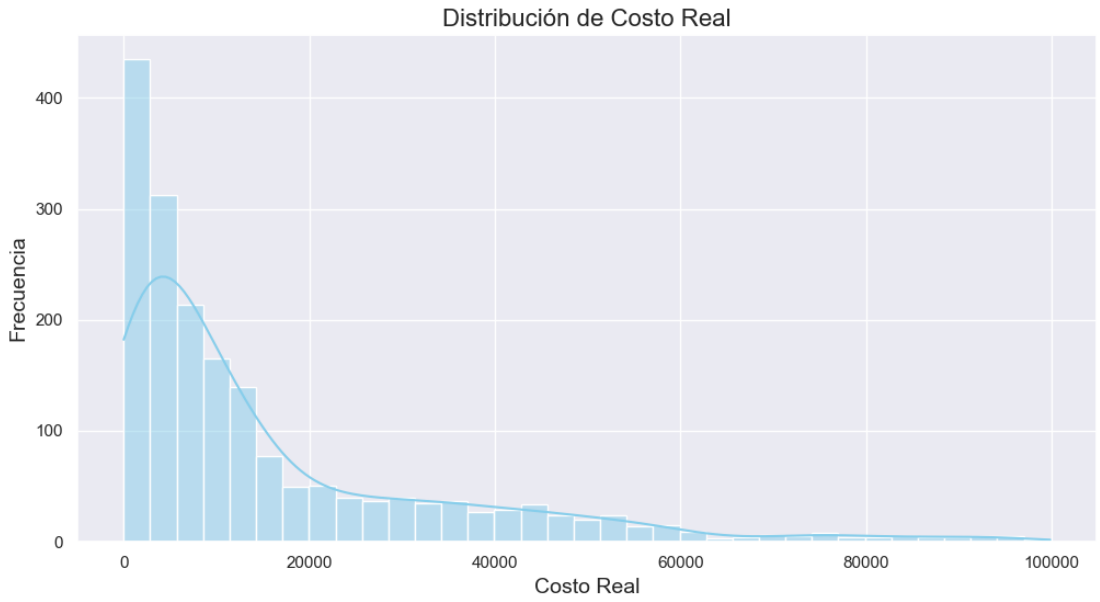


Figura 13: Distribución Costo Real Vehículos
Fuente: Elaboración propia

Además, es crucial considerar que este mapa representa el costo asociado al tiempo o al costo real de la simulación de evacuación, el cual varía en diferentes ocasiones. Las rutas calculadas sin considerar estas variables muestran una distribución diferente de tiempos. Al no existir conflictos de capacidad en las calles y, por ende, ninguna penalización de tiempo, las rutas y costos se verán afectados únicamente por la distancia. Al utilizar únicamente las rutas teóricas, se obtendrá un mapa diferente, como se observa a continuación, al calcular los costos asociados a cada ruta y, al igual que en el caso anterior, promediar los vehículos presentes dentro de un mismo nodo.

De todas formas gran parte de los vehículos no tendrán un costo mayor a $40{,}000$, aunque como se observa a continuación, los valores de la simulación con atascamiento generar una gran diferencia frente a una simulación sin ellos.

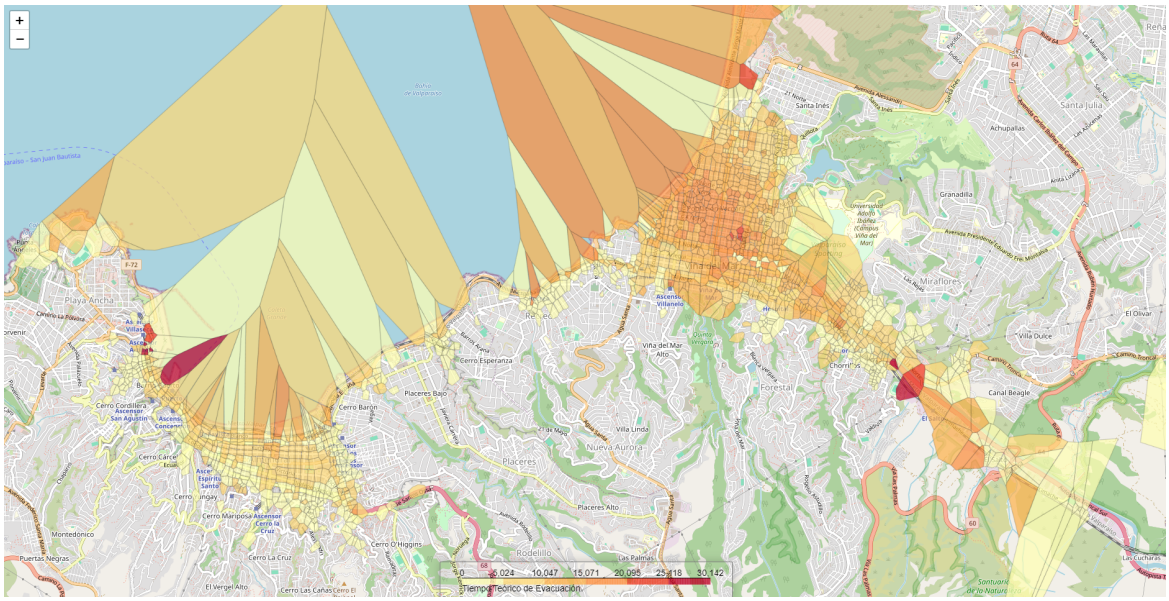


Figura 14: Costo de Evacuación Teórico, sin atascamientos
Fuente: Elaboración propia

Dentro del mapa de costos de rutas teóricas, sin tener en cuenta los conflictos de capacidad se observa que el centro de viña inicialmente no posee una concentración de altos costos en la zona anteriormente mencionada, sino que esta se encuentra con similares costos frente a las otras cuadras y aquella zona en general del plan de Viña del Mar, por lo que detectamos que dentro de esa zona los vehículos naturalmente debieron de sufrir gran cantidad de conflictos de capacidad a lo largo de sus rutas.

Por otro lado existen polígonos que nos presentar un rojo intenso, esto se debe a que se encuentran dentro de una carretera y por ende deben recorrer una larga calle sin posibilidad de desviar su ruta, si es que ellos siguen correctamente las reglas de transito, transcurriendo gran parte del camino en zona de peligro, mientras que nodos vecinos no se ven tan afectados debido a una mayor posibilidades de rutas a tomar.

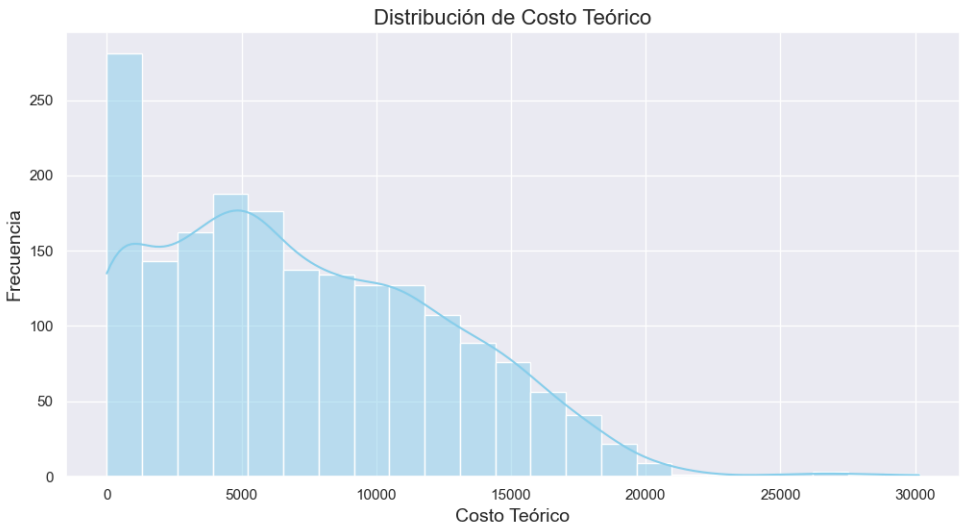


Figura 15: Distribución de Costo Teórico, sin atascamientos
Fuente: Elaboración propia

Claro está que la distribución de tiempos se vera disminuida, obteniendo la gran mayoría de costos en menos de $20{,}000$.

Dado esto se presenta la pregunta, ¿Cuáles fueron los nodos con una mayor variación de tiempos según el costo teórica de sus rutas?, siendo esta respuesta de gran utilidad para obtener las zonas que se verán enfrentadas a mayores atascamientos, y por ende poseer una mayor preocupación o prioridad frente a otras zonas, para obtener esta información basta con utilizar el costo real de evacuación y el costo teórico respectivo, visualizados anteriormente.

Para hacer un uso correcto de las variaciones y por ende mayor sufrimiento de atascamientos se debe realmente observar la proporción de estas dos métricas, relacionando las variables en una proporción $Costo\_Real/Costo\_Teorico$, ya que si se calcula la métrica de resta entre estos y se obtiene el costo agregado a la ruta, esta se vera sesgada por la distancia y costo mínimo que de por si nace geográficamente, por lo que al obtener la división podremos observar el incremento en base a su distancia respectiva. Teniendo en cuenta aquello y la metodología empleada, se obtiene el siguiente mapa según la proporcionalidad de cambio.

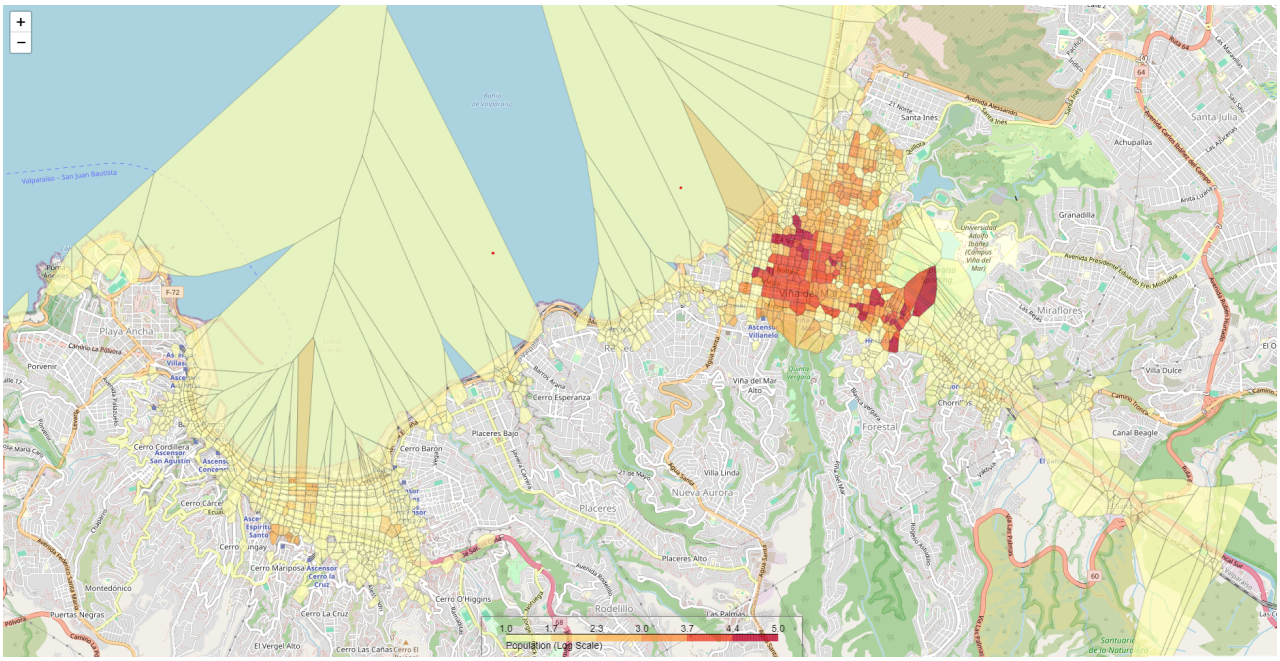


Figura 16: Diferencia Proporcional, Costo Real / Costo Teórico
Fuente: Elaboración propia

En el mapa 16 se observa que efectivamente la mayor diferencia de tiempos se encuentra en el centro de viña. donde haciendo un zoom hacia esa zona se observa en mayor detalle las calles, además de leves zonas naranjas en el centro de Valparaíso, el cual no fue detectable a simple vistan anteriormente.

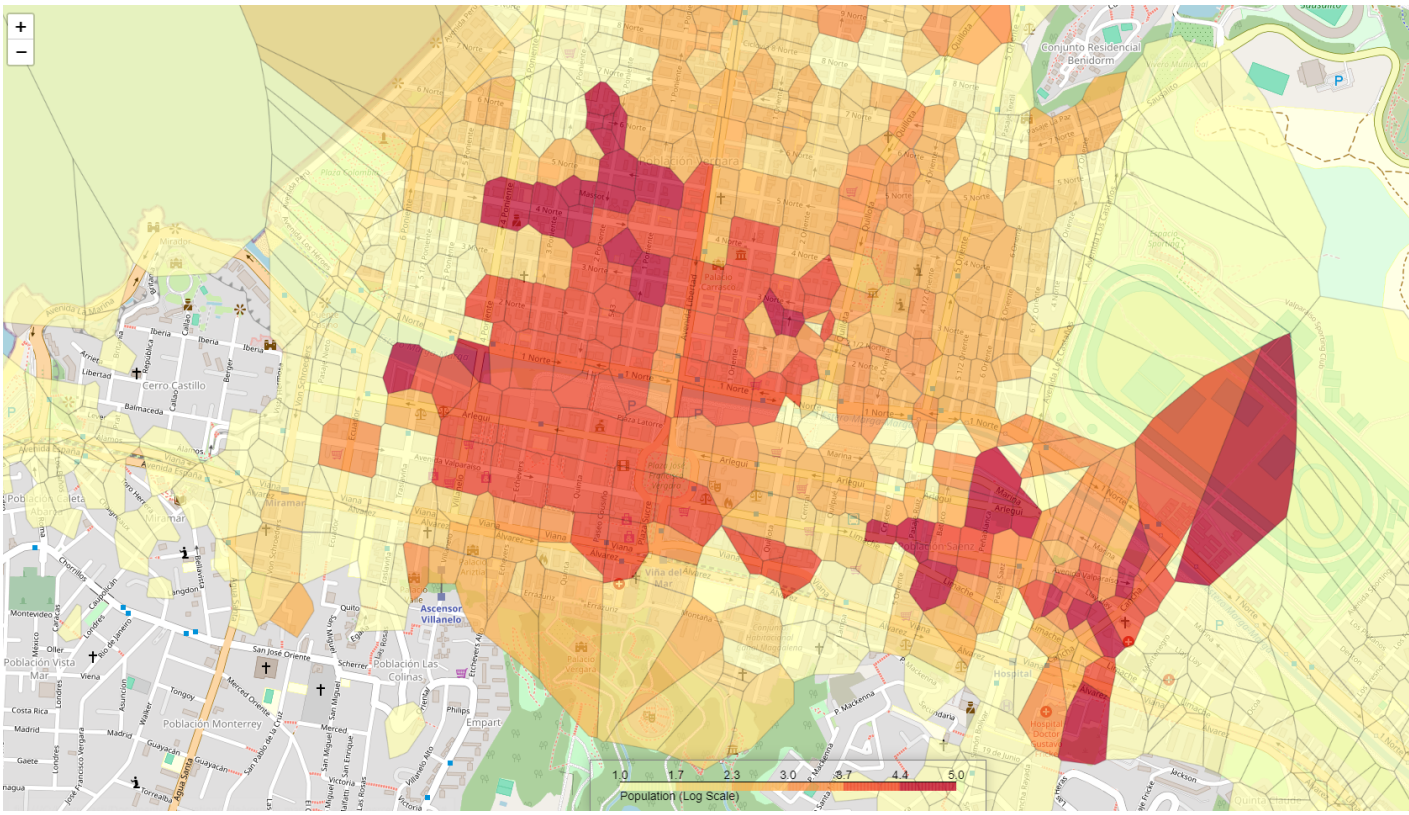


Figura 17: Zoom Diferencia Proporcional, Costo Real / Costo Teórico
Fuente: Elaboración propia

4.9. Mejoras a la red de Transporte

Si bien hasta ahora dio a conocer los costos que sufrieron los diferentes nodos o zonas geográficas utilizando la red de transporte con la mayor fidelidad posible a la actualmente utilizada y utilizando las rutas que entregaron un menor tiempo de evacuación total, aplicando el algoritmo iterativo presentado. En esta siguiente etapa se analizara en mayor profundidad el comportamiento y relevancia de los arcos o calles de la red, ya que si bien es importante utilizar un estudio de estrés de nuestra red bajo las leyes de transito del día a día y como serian utilizadas por la costumbre de los habitantes, a la hora de catástrofes también existen entidades como los carabineros capaces de llevar a una reorganización de ella y salir del marco establecido, al igual que las mismas personas tomaran la ruta que les parezca mas segura independiente de las direcciones establecidas. Por ejemplo se observa en una primera instancia cuales fueron las calles que presentaron un conflicto de capacidad dentro de la red, y cuales fueron las que no presentaron ningún conflicto a lo largo de la evacuación.

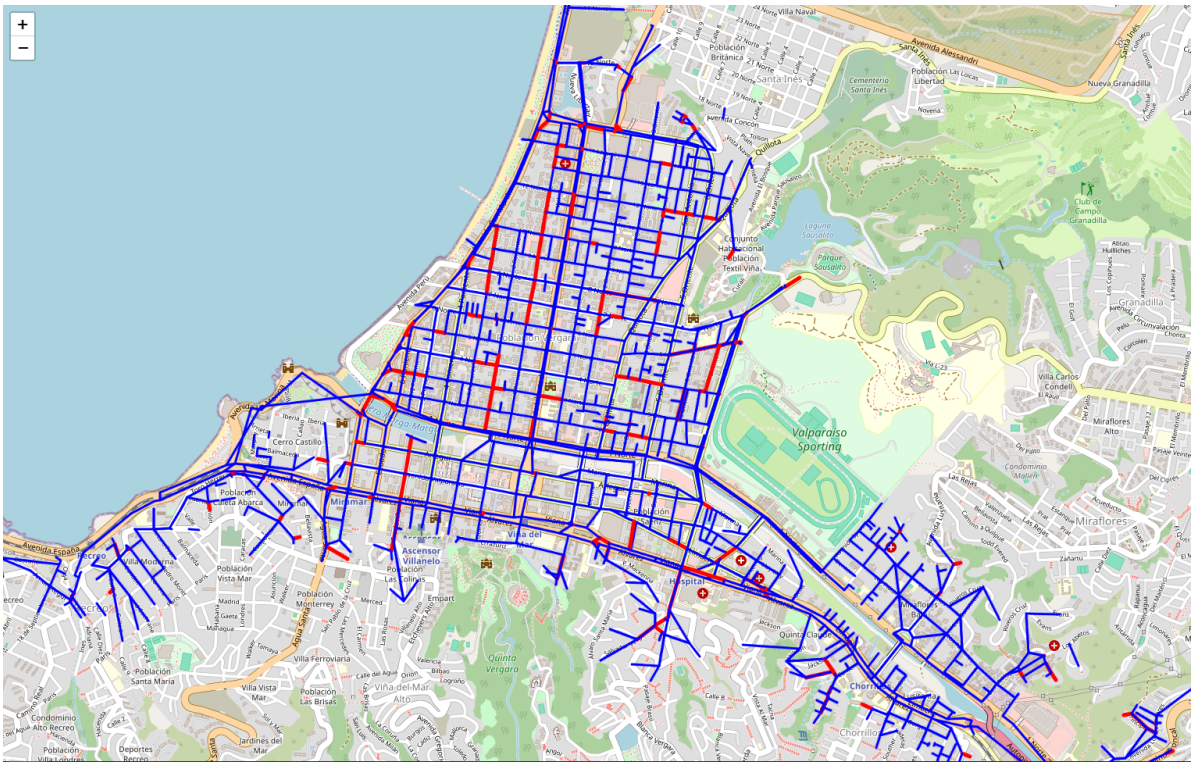


Figura 18: Calles con Conflictos de Capacidad (Rojo) Viña del Mar
Fuente: Elaboración propia

Dado en el mapa anterior del centro de Viña del Mar se observa cuales fueron las calles con conflictos de capacidad durante la evacuación óptima encontrada. Cabe destacar que dentro de la red de transporte existen 3443 arcos, de las cuales 284 generaron conflictos. Dado esto se rescata del mapa que existen dentro de las zonas residenciales del centro una concentración de estas calles, al igual que en los finales de las vías de escape considerando que existen calles que debido a su dirección no permiten generar la conexiones pertinente para evacuar la zona de peligro, dejando arcos sin utilizar y por ende generando una sobrecarga a la que si poseen una dirección hacia los cerros.

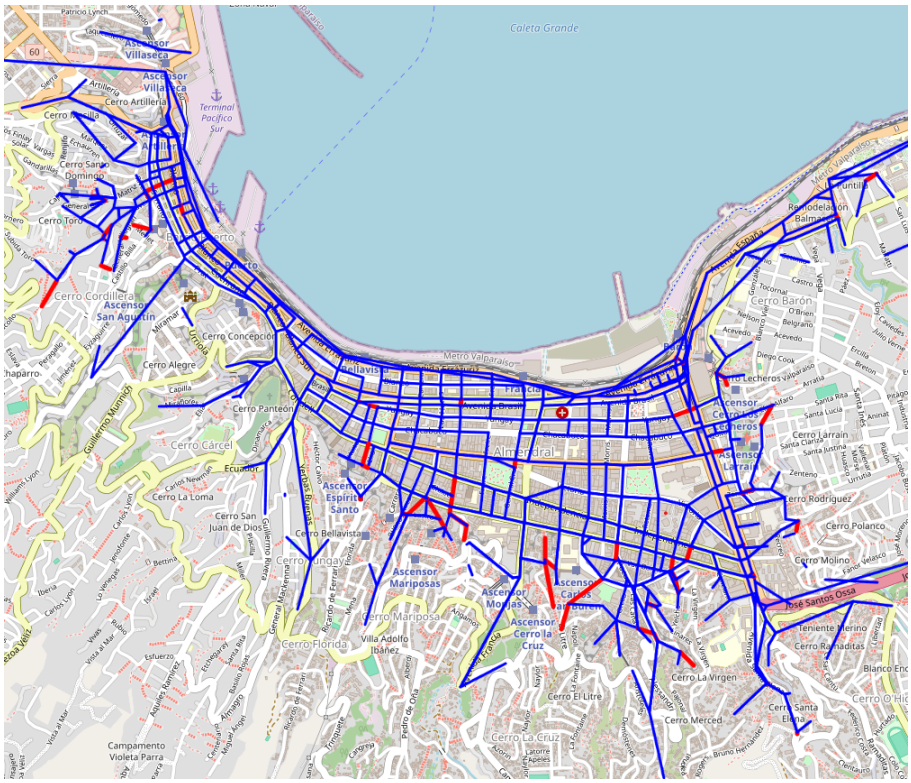


Figura 19: Calles con Conflictos de Capacidad (Rojo) Valparaíso
Fuente: Elaboración propia

La regla observada entre las dos comunas se mantiene, siendo así gran parte de la evacuación regida por los conflictos de capacidad a pesar de utilizar una metodología lineal al penalizar los tiempos de rutas, al igual que establecer que una mayor cantidad de calles con conflicto generara lógicamente un mayor tiempo de evacuación total, debido a que cada calle con conflictos tendrá por consecuencia el aumentos de los tiempos, siendo estas las responsables de aumentar los tiempos teóricos, y como ya vimos anteriormente pueden llegar a multiplicar cuatro veces el costo de ciertas zonas.

Por otro lado también es útil observar cuales fueron las calles de salida que presentaron un mayor trafico de vehículos utilizadas para la evacuación, con el fin de observar si es que existe alguna calle o zona con mayor relevancia o flujo de vehículos.

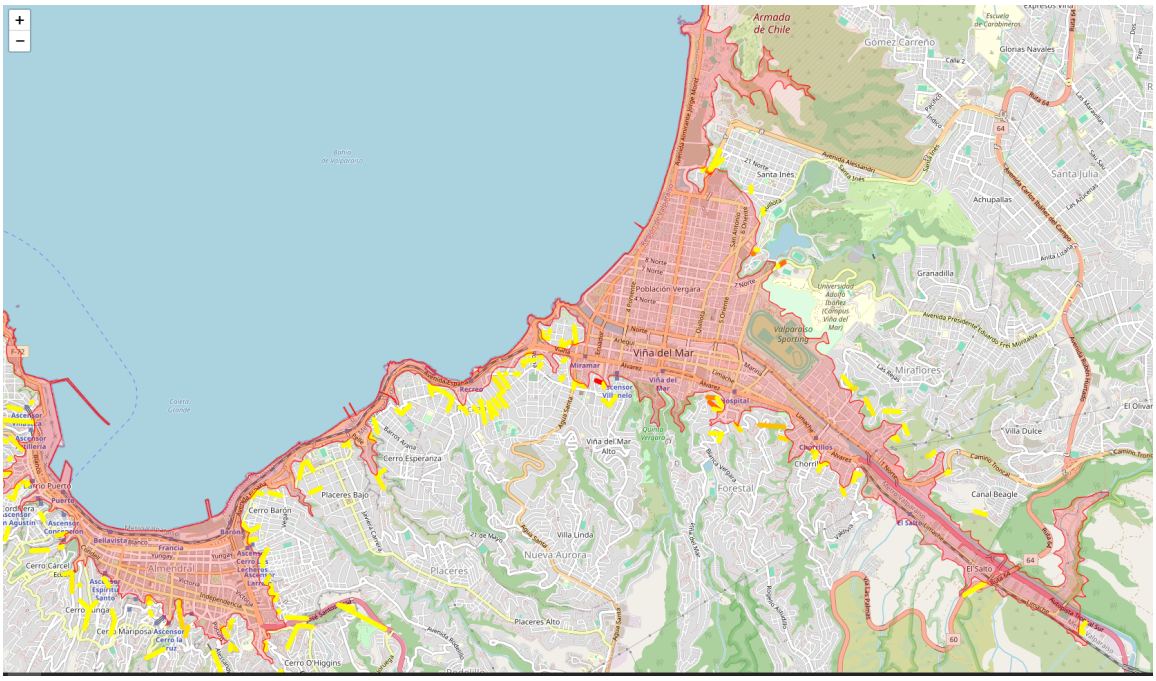


Figura 20: Utilización de Arcos
Fuente: Elaboración propia

Finalmente se observa cuales fueron las calles que obtuvieron un mayor flujo de vehículos a lo largo de la evacuación, siendo Translaviña una calle con flujo prioritario tanto como vía de evacuación como durante los inicios de la evacuación.

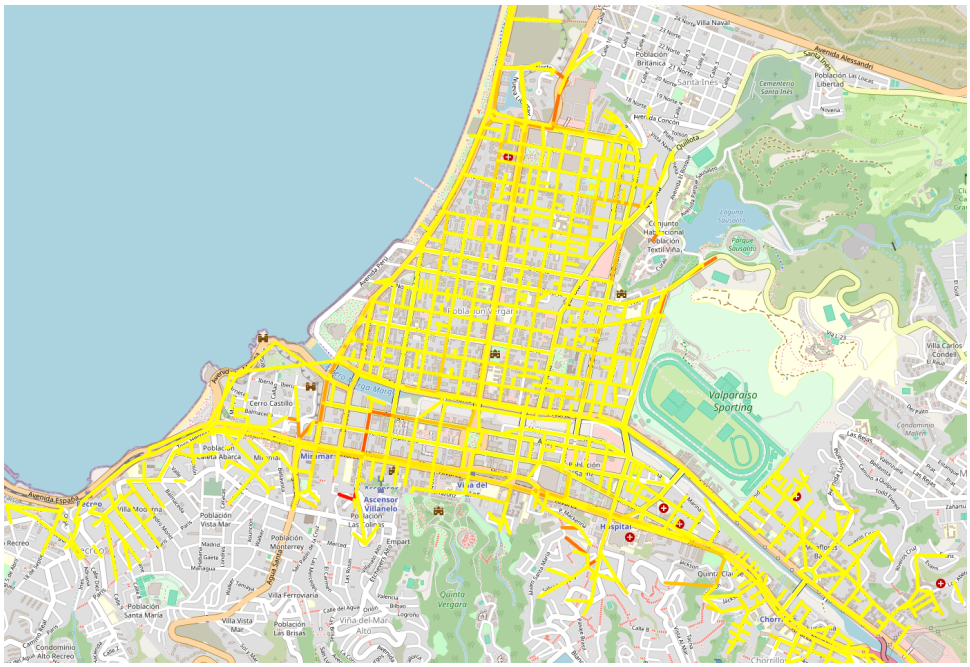


Figura 21: Cantidad de Vehículos por arco
Fuente: Elaboración propia

4.9.1. Modificación de Red de transporte

Anteriormente se observo un análisis exploratorio que logra resumir el comportamientos tanto de los vehículos como el de las calles para la red de transporte original, pero ahora se estudiara la modificación de la arquitectura de la red para el objetivo específico, la evacuación de tsnunamis, y comparar los cambios asociados y si es que existen algún comportamiento relevante a tener en cuenta.

Para obtener la red de transporte que mas se adecue al objetivo se utiliza nuevamente la metodología de algoritmos iterativa, *Hill Climbing*, del cual se enfocara principalmente en modificar las direcciones que posee la red de transporte en busca de minimizar el tiempo de evacuación. La distribución de población y por ende de vehículos es la misma que el paso anterior.

Para generar este nuevo algoritmo se poseen ciertas funciones reutilizadas del paso anteriores. Dada nuestra red de transporte (G), se modifican 10 arcos dentro de la zona de peligro y por ende de estudio, 5 asociados a los arcos bidireccionales y 5 asociados a los arcos unidireccionales, por cada iteración o movimiento, luego una vez generado el cambio se utiliza la nueva red candidata, generando el conjunto de rutas para los vehículos de forma *greedy*, asociando a cada vehículo su ruta de menor costo para evacuar la zona, luego utilizamos el algoritmo de simulación de evacuación usado anteriormente al igual que los mismos criterios y función objetivo del algoritmo iterativo, es decir, tiempo de evacuación y si es que

existe mismo tiempo se elige el de menor conflictos. Dado esto se obtuvo una nueva red especializada para el caso de estudio.

Para visualizar el cambio de red y las dirección que poseen se genero el siguiente mapa, el cual colorea los arcos según dos criterios, el primero es si es que apunta hacia el este u oeste, siendo el oeste en escala de rojo y el este en escala de verde, se definieron estas dos direcciones debido a la naturaleza de nuestra costa y que simplificando el problema, las calles buscan alejarse del mar, aunque como fue mencionado anteriormente y principalmente en las calles de Viana y Alvarez el movimiento es mas de norte o sur por lo pasajes conectados por estas avenidas. Luego según la intensidad de dirección, donde este y oeste es de mayor intensidad el color estos disminuyen según se acercan hacia el norte o sur. Por otro lado es importante considerar que existen calles con doble sentido, lo cual son consideradas como 2 arcos diferentes dentro de la red, y son coloreadas como azules ya que poseen ambas direcciones. Dado esto se tienen casos como Translaviña, las cuales podrían convertirse en un solo sentido, lo cual es considerado algo lógico según lo observado por los mapas, ya que posee una gran concurrencia de vehículos, conflictos de capacidad y es un punto de llegada activo. Esto es algo común dentro de la red de transporte que habita la zona de peligro teniendo 985, calles que poseen doble sentido por lo que existen 1970 arcos asociados a esos pares, dejando así 1674 arcos que utilizan una dirección única para conectar dos nodos, sumando así los 3644 presentes en la zona.

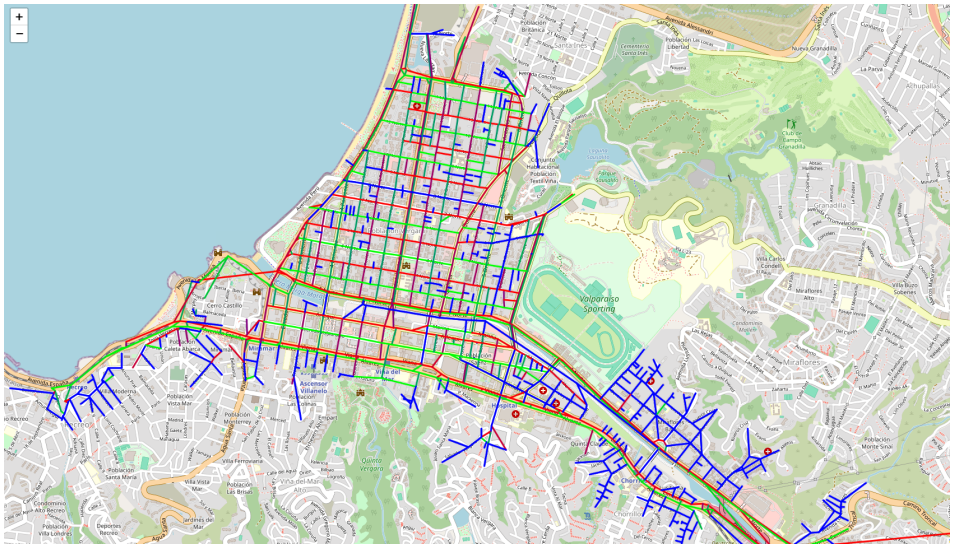


Figura 22: Red Original
Fuente: Elaboración propia

Dado los resultados obtenidos inicialmente utilizando la red original y su solución greedy con un costo de 95105 de evacuación, mientras que utilizando este algoritmo iterativo de modificación de red obtuvo un costo 57145 utilizando únicamente la solución greedy, por lo que se observan mejoras significativas utilizando cambios de red frente a las rutas de vehículos. Esta diferencia de escala en las mejoras asociadas al costo da a luz sobre las influencias que

pueden llegar a tener los diferentes aspectos de nuestra evacuación, generando una mejora horizontal al modificar la calle y al organizar los vehículos o aumentar las capacidades de la calle se podría considerar una mejora vertical.

El Algoritmo utilizado y las dependencias necesarias se presentan en el siguiente cuadro. Este posee las mismas características frente al algoritmo anterior dado la estructura general de un *Hill Climbing*, siendo este mas simple en su movimiento y la reutilización de las funciones ya generadas. Además las diferentes redes se guardan en un archivo *.pkl* utilizando la librería *Pickle*, con el fin de generar diferentes procesos que comparten esta carpeta con las mejores soluciones, de tal forma de aumentar el gradiente del algoritmo y utilizar en mayor medida los recursos computacionales al utilizar múltiples procesos. Otra mejora fue utilizar *threads*, para la generación de cada ruta debido a que esta al ser *greedy* simplemente busca la ruta de menor costo para cada nodo, siendo este un proceso independiente.

Algoritmo 13 Proceso de optimización por generaciones

Require: $num\_of\_generations$: Número de generaciones
Require: $Candidate\_G$: Grafo candidato
Require: $zona\_de\_peligro$: Conjunto de polígonos de zona de peligro
Require: $num\_samples$: Número de muestras a seleccionar
Require: $edges\_inside$: Conjunto de arcos dentro de polígonos
Require: $cars\_count\_by\_node$: Conteo de autos por nodo
Require: $vehicle\_seating\_capacity$: Capacidad de asientos del vehículo
Require: $directory\_path$: Directorio para guardar los resultados
Require: $leaderboard\_time$: Tabla de tiempos de evacuación

for $generation = 0$ **to** $num\_of\_generations - 1$ **do**
  $G\_cambios, success, edges\_inside\_of\_set, path\_costs \leftarrow$ ModificarRed$(Candidate_G, zona_de_peligro, num_samples, edges_inside_set)$
  $Candidate\_G \leftarrow$ copy$(G_cambios)$
  $routes\_cars\_original, movimientos\_realizar, seating\_capacity\_car \leftarrow$ generate_routes_cars_fast$(car_count_node, nodes_inside, path_costs, vehicle_seating_capacity)$
  $EvacuationTime, conflict, conflictos\_car, finished\_cars, tiempo\_finishes\_car \leftarrow$ generate_simulation$(Candidate_G, routes_cars_original)$
  open$((EvacuationTime) + ($conflict$) + ($generation$) +' .pkl'$
  pickle.dump$(leaderboard_time[EvacuationTime], file)$
  $[Candidate\_G, best\_conflict, 2\_ways\_edge, one\_ways\_edge, edges\_inside], lowest\_n1 \leftarrow$ get_values_from_lowest_Time
  $leaderboard\_time[lowest\_n1] \leftarrow [Candidate\_G, conflict, two\_ways\_edges, one\_ways\_edges, edges\_inside\_of\_set]$
  $best\_case \leftarrow$ sorted$($leaderboard_time.keys$())[0]$
end for

La función encargada de hacer modificaciones a la red, al poseer una posibilidad de al hacer los cambios, generar nodos sin posibilidad de utilizar una ruta apta para evacuar la zona de peligro, rompiendo así la eficacia de la red, se verifica si es que la función que genera los caminos no falla, y hasta que se generan cambios aptos para todos los vehículos no retorna la nueva red, al igual que los datos asociados a la red, tales como, cantidad de vías dobles y simples.

Algoritmo 14 ModificarRed

Require: G: Grafo de NetworkX que representa la red vial.
Require: $zona\_de\_peligro$: Conjunto de polígonos que representan las zonas de peligro.
Require: $num\_samples$: Número de bordes a modificar en cada intento.
Require: $max\_attempts$: Número máximo de intentos para realizar cambios.

$success \leftarrow$ False
while $attempts < max\_attempts$ y not success **do**
  $G\_cambios \leftarrow$ copy.deepcopy(G)
  $two\_ways\_edges, one\_ways\_edges, edges\_inside\_polygons\_of\_set \leftarrow$ identify_edges_inside_polygons$(G_cambios, zona_de_peligro)$
  $edges\_for\_changes\_doble \leftarrow$ random.choices(list$(two_ways_edges), k = num_samples)$
  $edges\_for\_changes\_one\_way \leftarrow$ random.choices(list$(one_ways_edges), k = num_samples)$
  for id_from, id_to en $edges\_for\_changes\_doble$ **do**
    $capacity \leftarrow 2 \times G\_cambios.edges[id\_from, id\_to]['capacity']$
    $distancia \leftarrow G\_cambios.edges[id\_from, id\_to]['distancia']$
    $G\_cambios.remove\_edge(id\_from, id\_to)$
    $G\_cambios.add\_edge(id\_to, id\_from, capacity, distancia, 0)$
  end for
  for id_from, id_to en $edges\_for\_changes\_one\_way$ **do**
    $capacity \leftarrow$ int$(G_cambios.edges[id_from, id_to]['capacity']/2)$
    $distancia \leftarrow G\_cambios.edges[id\_from, id\_to]['distancia']$
    $G\_cambios.remove\_edge(id\_from, id\_to)$
    $G\_cambios.add\_edge(id\_to, id\_from, capacity, distancia, 0)$
    $G\_cambios.add\_edge(id\_from, id\_to, capacity, distancia, 0)$
  end for
  $path\_costs, success \leftarrow$ generate_shortest_routes$(G_cambios, edges_inside_set)$
end while
return $G\_cambios, success$

Los resultados de este algoritmo y la especialización de la red genero una mejora modificando una gran variedad de arcos, donde si hacemos una diferencia simétrica de los conjuntos de arcos, obtenemos 1201 arcos diferentes, lo cual genera una mejora significativa obteniendo cerca de la mitad del tiempo de evacuación como mencionamos anteriormente, modificando así de dirección los siguientes arcos presentes en el mapa 23.

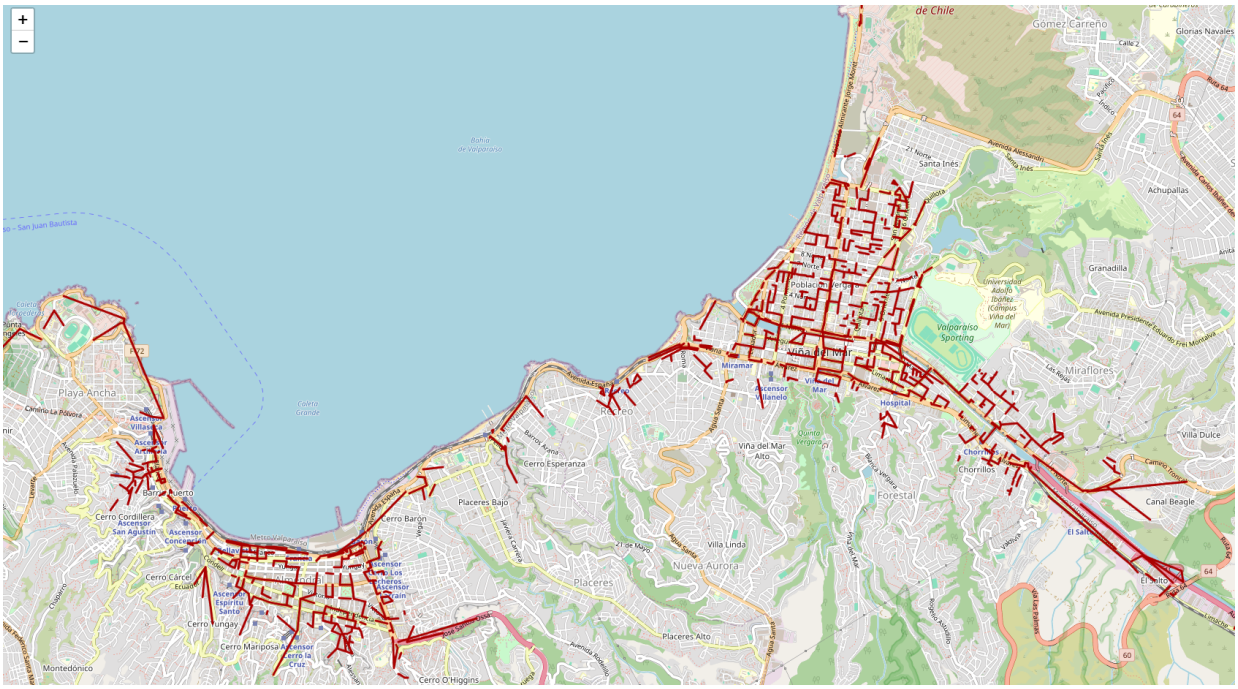


Figura 23: Arcos Modificados
Fuente: Elaboración propia

Prestando atención en la zona de mayor interés debido a poseer altos niveles de conflicto y ser participes de mayores tiempo de evacuación al igual de presencia demográfica, se observa el centro de Viña del Mar con mayor aumento en el mapa.

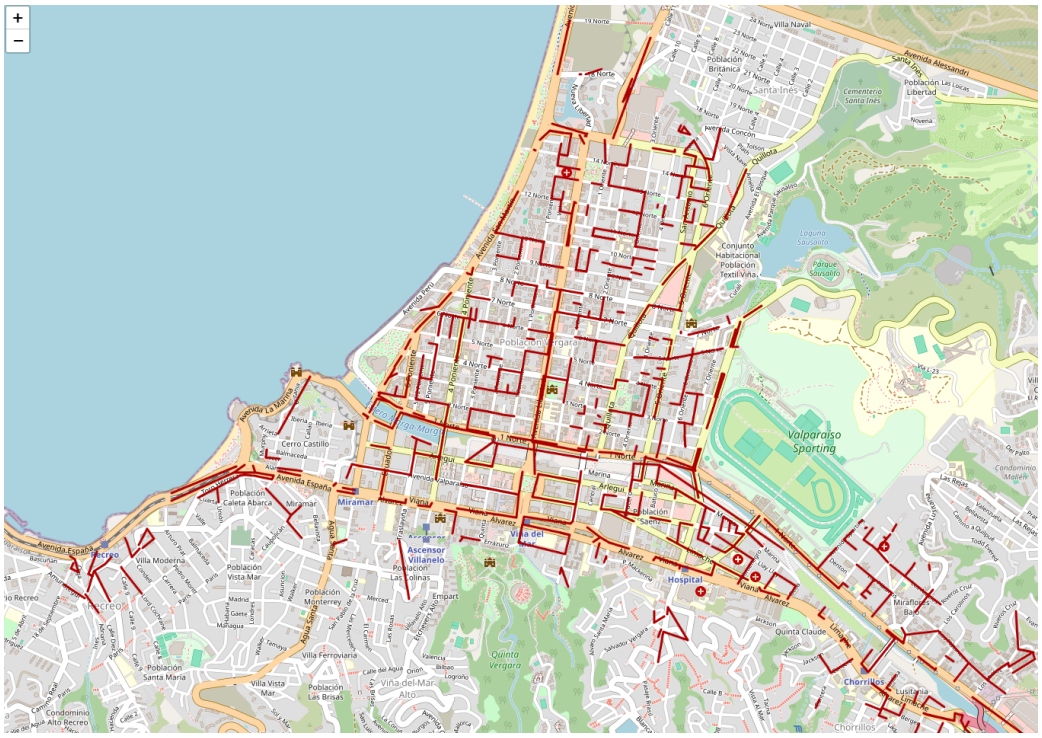


Figura 24: Arcos Modificados Viña del Mar
Fuente: Elaboración propia

Obteniendo en esta zona gran cantidad de cambios, pero estos cambios fueron útiles para obtener rutas de menor costo para minimizar tiempos o fueron modificadas con el fin de minimizar conflictos, bueno, no necesariamente fueron por una única corriente en particular sino que una cooperación entre estas dos, como son capaces los algoritmos iterativos generando mejoras en base a la función objetivo y la libertad presentada por los movimientos y el azar de ellas para buscar entre todo el espacio de búsqueda, el cual debemos destacar que existe una menor cantidad de redes posibles, teniendo los nodos fijos, frente a la cantidad de rutas posibles en aproximadamente *200.000* vehículos.

Para obtener de mejor forma el criterio presentado por esta nueva red optimizada se presentan los mapas anteriormente generados en el paso de optimización de rutas. A continuación se visualizan las direcciones de los arcos.

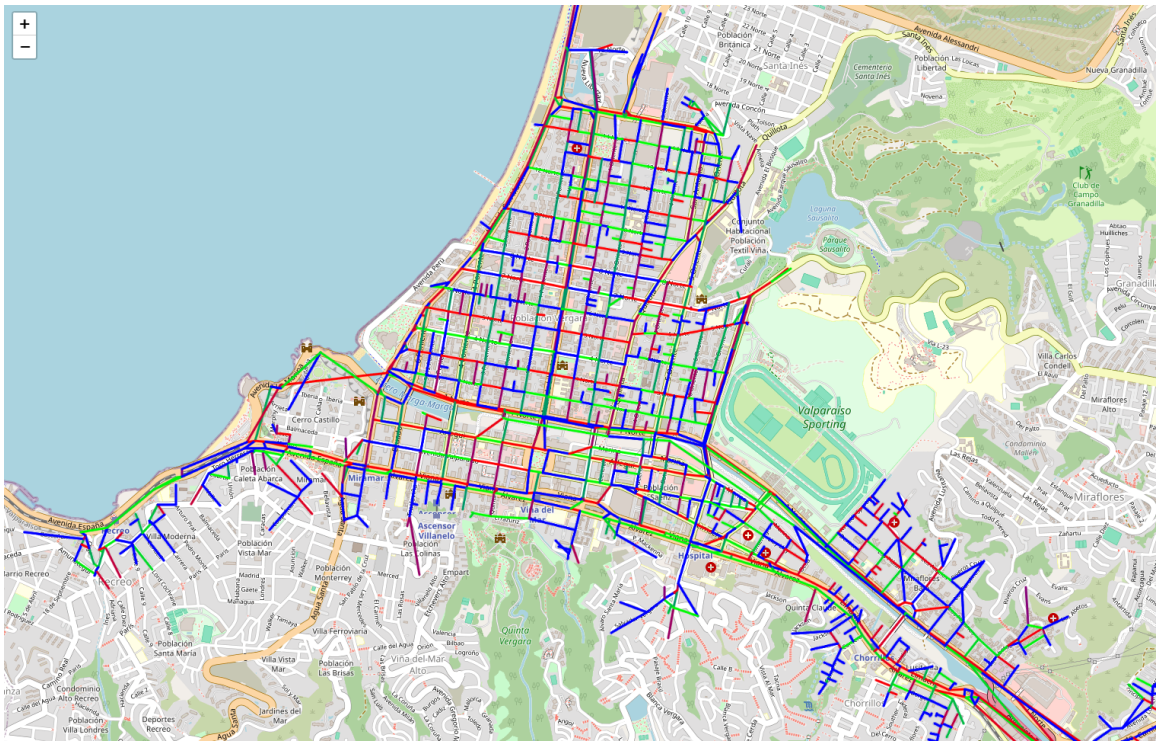


Figura 25: Red Optimizada
Fuente: Elaboración propia

Lo primero que se logra observar es que existe un menor orden natural para las calles, como por ejemplo no se observa que Viana y Alvarez poseen una misma dirección a lo largo de ella, al igual que Arlegui y calle Valparaíso, sin contar los cambios realizados en el centro asociados a las calles conectadas a avenida Libertad y sus alrededores, generando una red a la vista mas caótica y con mayor complejidad, algo que no necesariamente genera una utilidad o naturalidad frente a la evacuación, pero que si entrega cuales son las ventajas de esta nueva red.

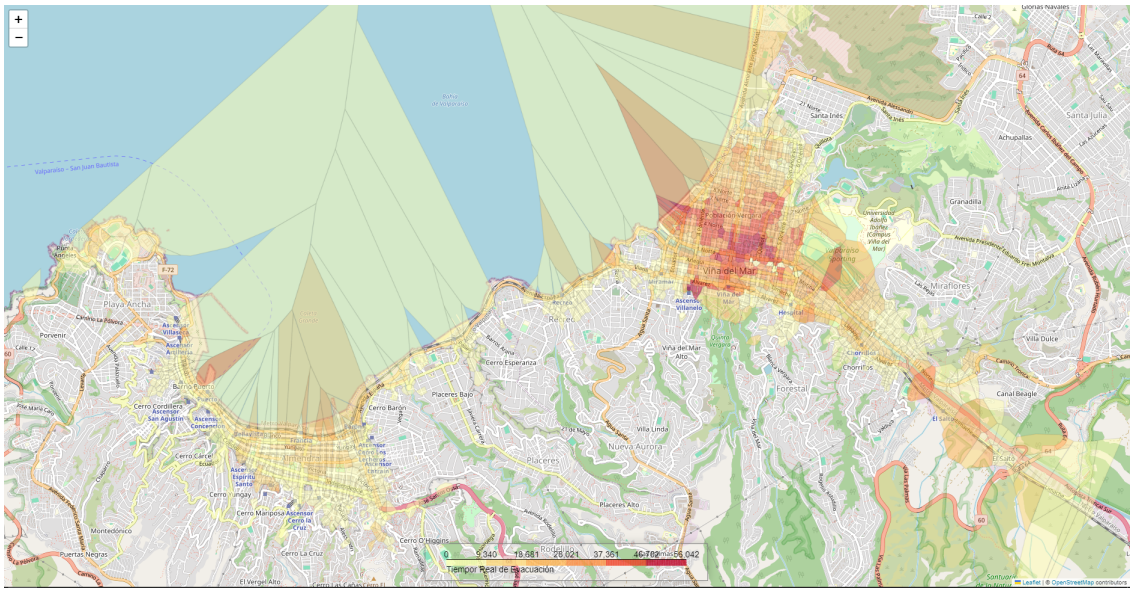


Figura 26: Costo de Evacuación Real, Red Optimizada
Fuente: Elaboración propia

Nuevamente las regla es clara entre los costos de evacuación entre las dos comunas, pero se observa que el sector donde los costos mas concentrados se encuentran a aumentando de tamaño generando así una áreas de tiempo mas uniforme, mientras que en Valparaíso no se demuestra una diferencia significativa.

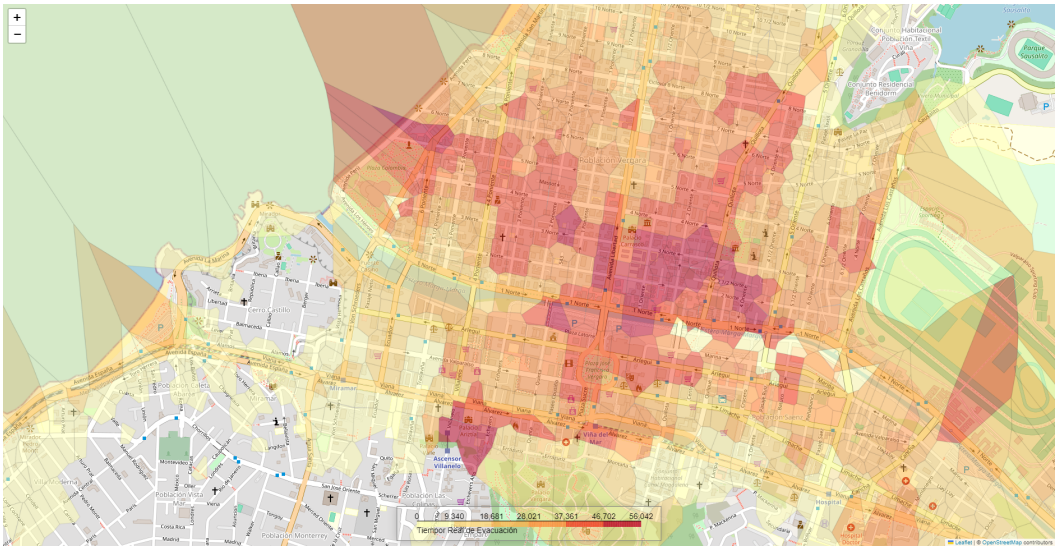


Figura 27: Costo de Evacuación Real Zoom, Red Optimizada
Fuente: Elaboración propia

En el zoom de 27 se ve mas clara la distribución de área de costos, teniendo como salvedad

frente al anterior gráfico de tiempos 12 la concentración de mayor tiempo se encuentra hacia el este de la avenida Libertad, a pesar de que esta área se encuentra mas cerca de la zona segura, esto probablemente fue un sacrifico realizado por el algoritmo con el fin de facilitar y priorizar las zonas de mayor peligro originalmente, aunque igual se observa cerca del hotel O'Higgins, siendo este un resultado mas lógico. Por otro lado llama la atención que el sector de Villanelo se encuentra entre los mayor a evacuar, debido a que ahora debe avanzar una mayor cantidad de tiempo siguiendo la avenida de Alvarez como menor costo.

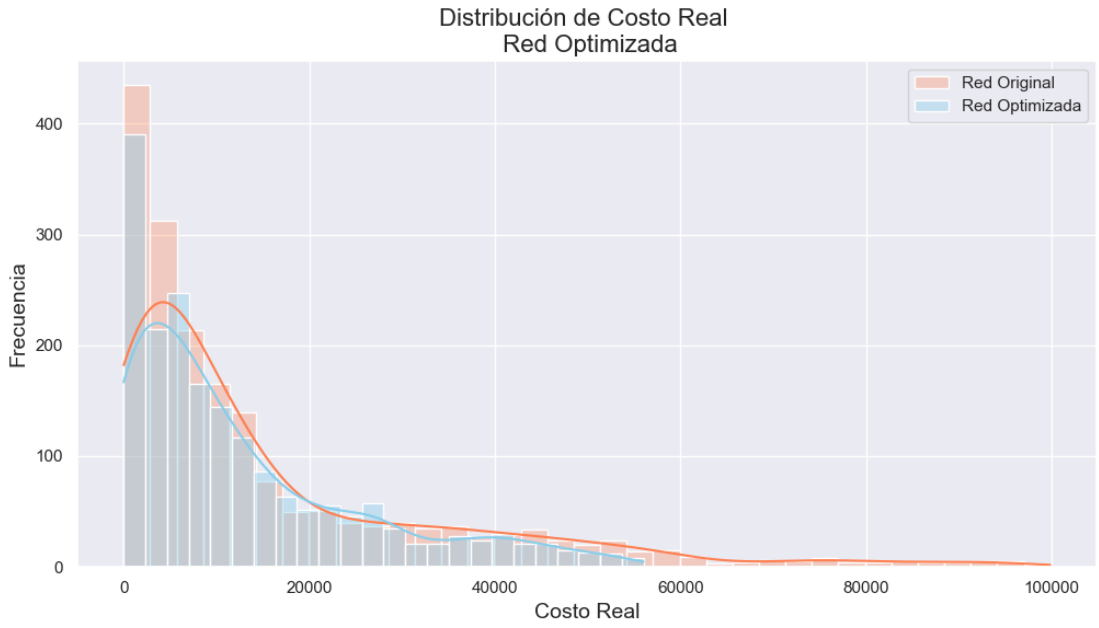


Figura 28: Distribución Costo de Evacuación Real Zoom, Red Optimizada
Fuente: Elaboración propia

La distribución de tiempo es similar, pero el tiempo máximo es menor en la red optimizada debido al fin que tiene esta, además dentro del rango de $[20{,}000 : 30{,}000]$ es mayor la red optimizada, debido a que al tener menor tiempo máximo de evacuación se ven obligadamente a aumentar tiempos en otros sectores, además de perjudicar a vehículos que ya poseían menores tiempos con el fin de minimizar el tiempo total, algo interesante de realizar ya que nos da a la luz los recursos limitados que posee una red, siendo obligada a generar *trade-off* o compensaciones entre los tiempos de los vehículos, por el fin de la función objetivo al minimizar el tiempo máximo de la evacuación.

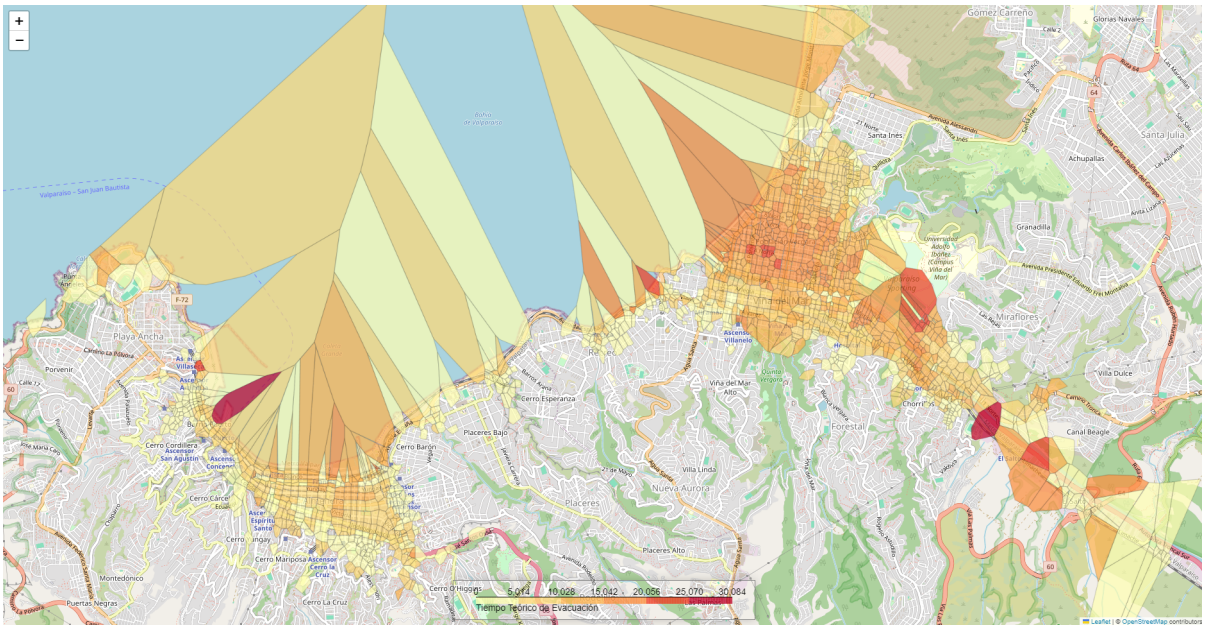


Figura 29: Costo de Evacuación Teórico, Red Optimizada
Fuente: Elaboración propia

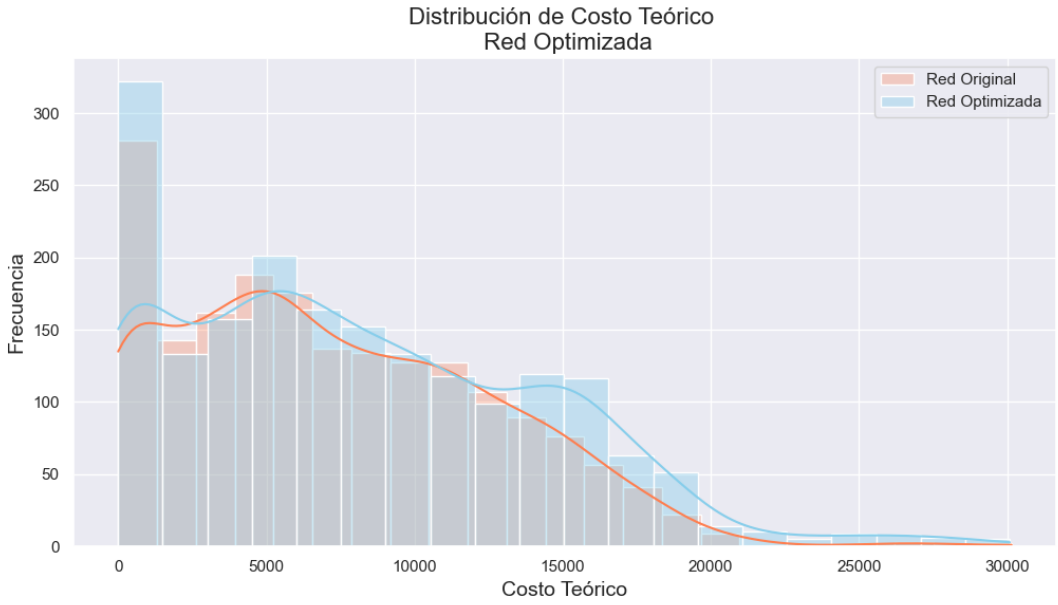


Figura 30: Distribución Costo de Evacuación Teórico
Fuente: Elaboración propia

A continuación se observa que los tiempos teóricos asociados a la red optimizada son mayores frente a la red original, pero a pesar de estos los tiempos reales y el costo de la evacuación

total es menor, por lo que sin duda alguna se tiene que los tiempos agregados por calles con conflictos de capacidad deben de ser menor, por lo que dado la pregunta anterior de si es que se generaban rutas con menor distancia a recorrer o si es que se optimizaban la cantidad de conflictos de capacidad, se ve la respuesta con el anterior gráfico 30, respondiendo que la nueva red disminuye las penalizaciones de costo, este análisis también es colaborado, en el gráfico de proporciones entre costo real y costo teórico.

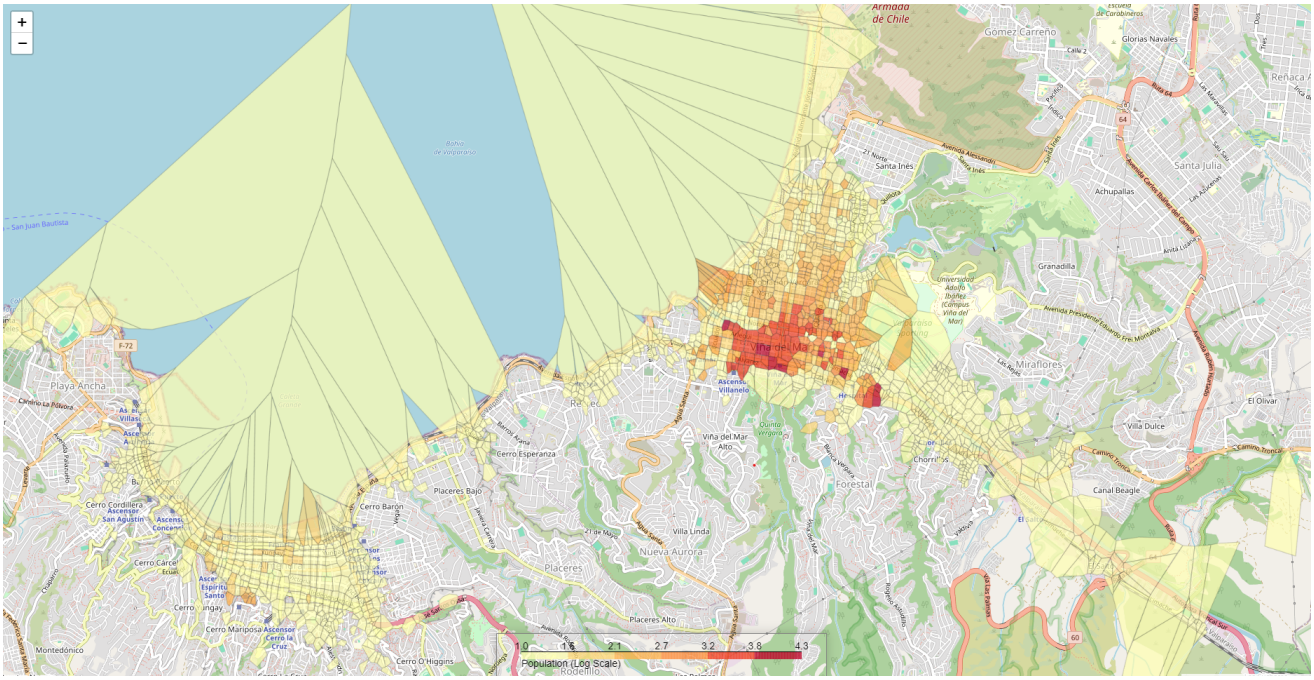


Figura 31: Costo de Evacuación Proporcional
Fuente: Elaboración propia

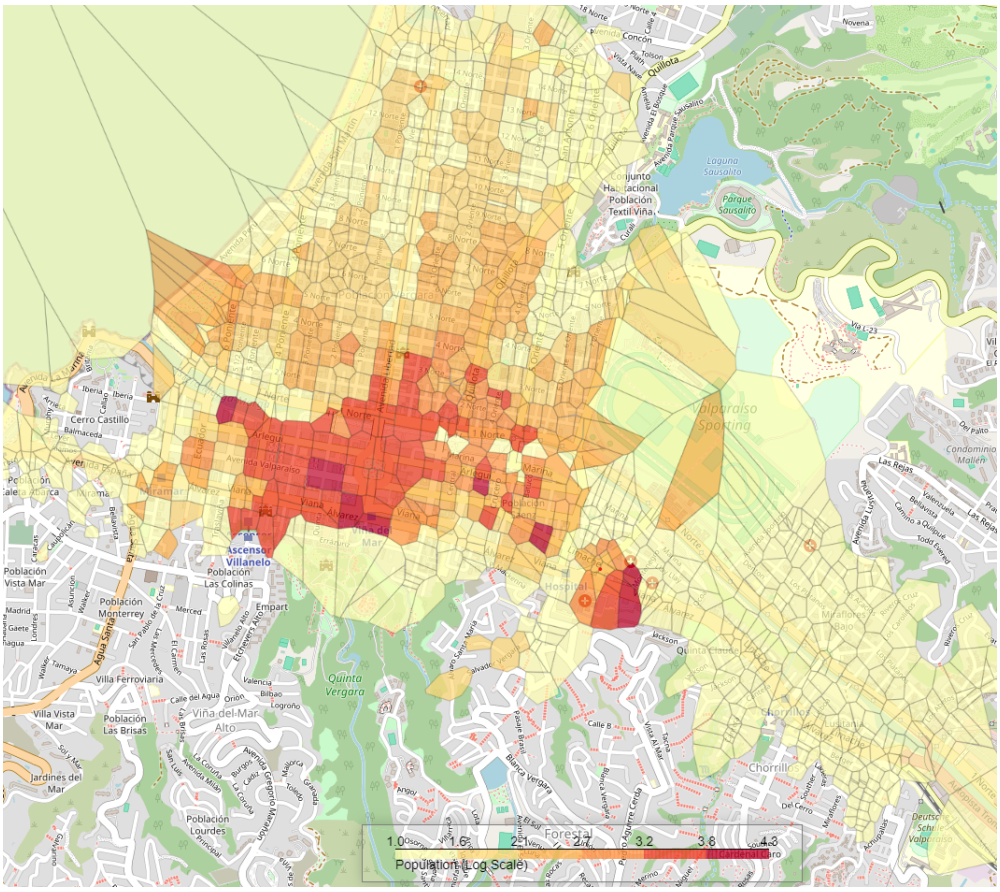


Figura 32: Costo de Evacuación Proporcional Zoom Viña del Mar
Fuente: Elaboración propia

EL mapa de comparaciones proporcionales entre los dos tiempos entrega similar distribución que los costos reales, obteniendo la mayor concentración de penalizaciones en calle Valparaíso en el sector de plaza Vergara y toda esa zona antes del puente, mientras que el área de costos reales también considera el sector al otro lado del puente, ya que este de por si se encuentra más en el centro de la zona de peligro.

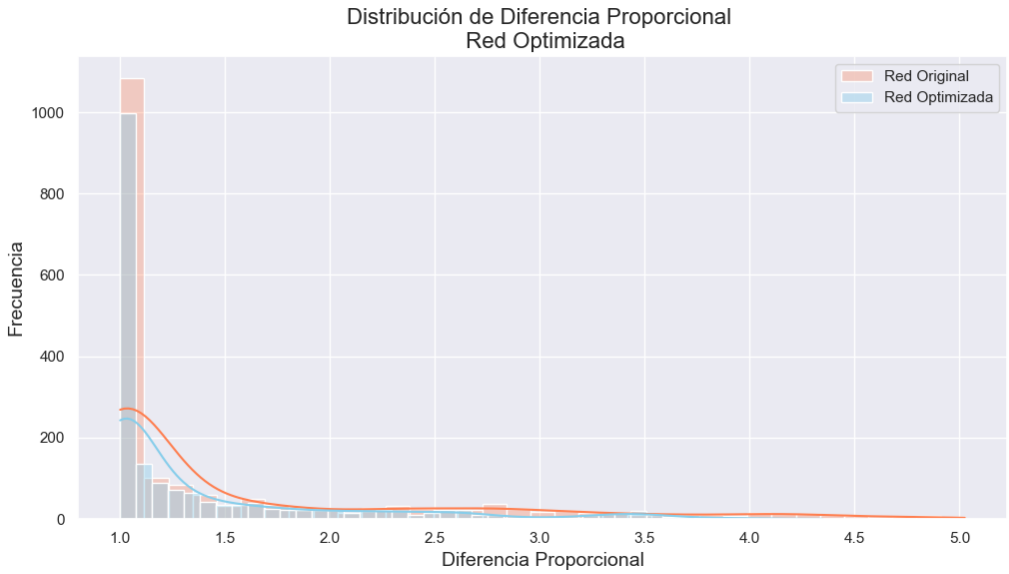


Figura 33: Distribución Costo de Evacuación Teórico Zoom
Fuente: Elaboración propia

Claramente se mantiene la relación pero las proporciones son menores en la red optimizada, donde cabe considerar que estas rutas no se encuentran optimizadas por el algoritmo iterativo que modifica las rutas generadas de formar *greedy*, dejando a la vista una ventana de mejora y una mayor comparación.

Por ultimo se obtienen los mapas asociados a los arcos de salida con mayor utilización al igual que las calles con mayor flojo de vehículos, presentes a continuación.

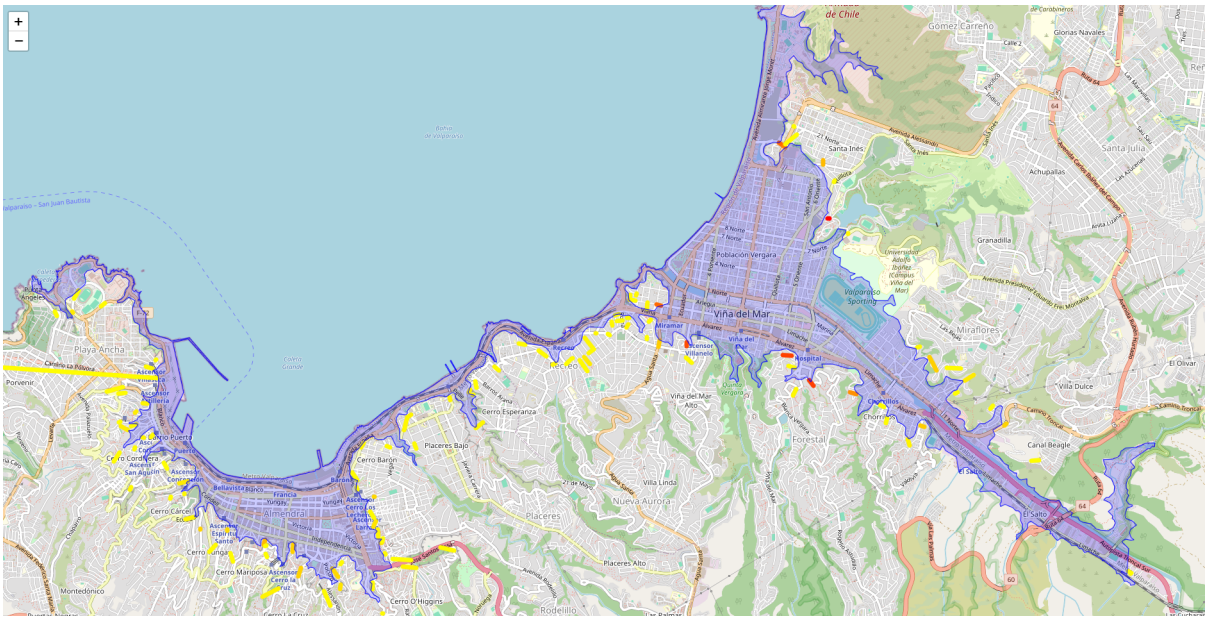


Figura 34: Calles de Salida Red Optimizada
Fuente: Elaboración propia

La diferencia de rutas no es muy diferente a la red anterior pero esta vez podemos se observa una mayor distribución entre las rutas con mayor color rojo, obteniendo nuevamente la importancia del Cerro Castillo en la calle saliente de calle valparaíso al igual que las calles asociadas a Translaviña.

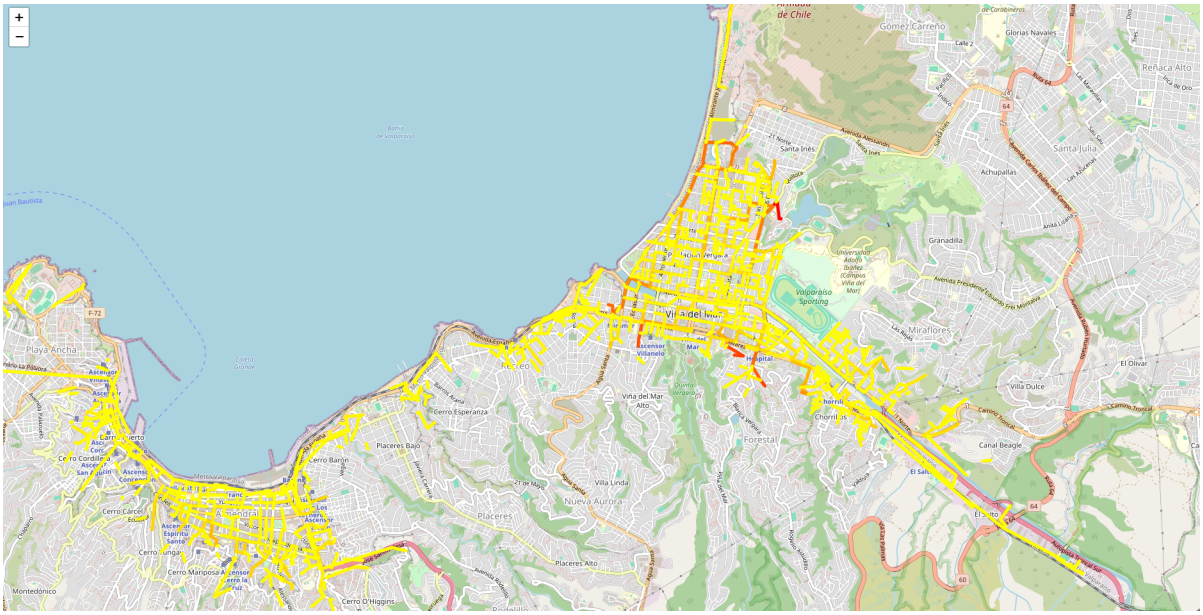


Figura 35: Utilización de Calles Red Optimizada
Fuente: Elaboración propia

Las calles en uso se ven principalmente condensadas de mayor forma en Viña del Mar.

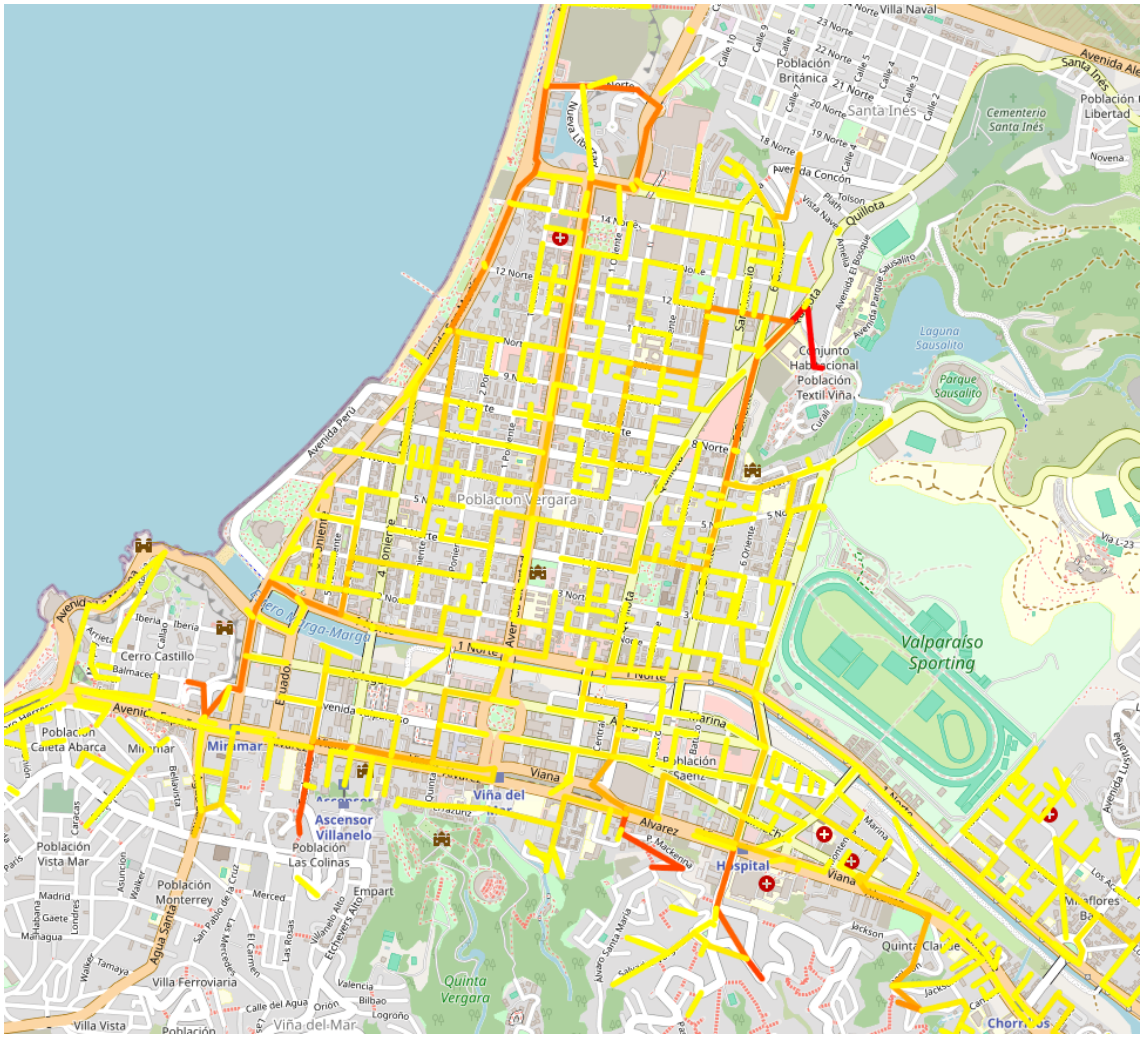


Figura 36: Utilización de Calles Red Optimizada Zoom Viña del Mar
Fuente: Elaboración propia

Por lo que haciendo un zoom podemos observar nuevamente que las calles mas utilizadas se encuentran en las salidas, al igual que nuestra nueva red observar una mayor dispersión de sus vehículos, utilizando de mejor forma y con menores conflictos las diferentes calles.

CAPÍTULO 5
CONCLUSIONES

La representaciones de fenómenos de la vida natural es sin duda un desafió complejo de abarcar, debido en una primera instancia al presentarse el factor humano dentro de ella y la gran cantidad de variables que tenemos en cuenta al tomar nuestra decisiones y siendo estas aun mas complejas al vernos en una situación de riesgo, por lo que es de vital importancia mantener informados sobre los procesos a seguir de forma que estos se encuentren dentro de la comunidad naturalmente y otorgar así las herramientas suficientes para poder reaccionar de forma informada y siguiendo un protocolo establecido con el fin de facilitar la toma de decisión y generar un proceso que se desarrolle en conjunto, ya que la evacuación y utilización de la red de transporte es un sistema caótico donde las acciones llevadas a cabo en un punto afectaran a otro punto lejano de la red en un tiempo diferente.

Por otro lado el representar una red de transporte utilizando datos que clasifican las calles y su capacidad utilizando datos categóricos, generados en base a la diversa cantidad de calles presentes en diferentes ciudades del mundo, es una buena aproximación pero igualmente esta posee un rango de mejora al utilizar imágenes satélitales, siendo este el principal factor de mejora para un futuro, debido a que utilizando esta información visual y en conjunto con técnicas de *computer vision* se puede representar la red de transporte con una mayor fidelidad, al otorgarnos datos útiles y así obtener la capacidades correspondientes, observando empíricamente tanto el largo como el ancho de la calle, además del estado actual de aquella. Por otro lado al poder observar estas calles y obtener imágenes de forma constante y con poco tiempo entre aquellas se puede predecir el trafico y distribución de vehículos al gusto de uno, ya sea en diferentes horarios y épocas del año, sin tener en consideración el alto costo que actualmente tienen estas imágenes, claro esta. Además ya avanzando cronológicamente en los capítulos de esta memoria también el espacio que abarcan los diferentes vehículos y clases que existan dentro de una calle, como se intento realizar con las micros y sus recorridos, debido a que se poseía información capaz para aquello, llevando a que estas clases pasen a ser una distribución y la medición precisa de los tamaños y espacio vació entre estos vehículos.

A pesar de aquello igual existe información capaz de generar abstracción y poder representar con datos cuantitativos la realidad de la región como se realizó en el transcurso de esta memoria, por lo que considerando las diferentes fidelidad que se pueden llegar a abarcar esto es únicamente un paso independiente frente a la simulación de una evacuación, donde se utilizo una representación lineal de las penalizaciones de tiempo frente a un atascamiento, siendo esta capaz de presentar jerarquías dentro de las calles y la simulación para identificar puntos débiles o desde un diferente punto de vista la relevancia que presenta en la red, ya que independiente de aquello se logra generar el conjunto de calles que poseen una mayor participación y uso dada nuestra distribución poblacional y zona de peligro.

El algoritmo utilizado para simular la ecuación es una pieza fundamental del estudio. En una primera instancia ya que el *output* de el refleja los resultados para el próximo estudio y además presenta la relación de todas las representaciones anteriormente formuladas, por lo que es clave realizar esfuerzos extras para realizar esta función de la forma mas eficiente posible, ya que todo este tiempo adicional generado tendrá finalmente un retorno de tiempo, debido a que esta es ejecutada miles de veces a lo largo del desarrollo de resultados y las mismas pruebas realizadas, por lo que independiente de la representación de solución realizada al estudiar la red de transporte se debe tener un especial énfasis en el.

Por otro lado los tiempos desarrollados no se deben considerar de manera fiel debido a que se utilizaron distancias y costos, además de no tener como consideración el tiempo asociado de las personas en subirse al vehículo, y por otro lado los tiempos asociados a semáforos tampoco se toman en cuenta, sino que el principal alcance de esta memoria es un análisis de estrés de la red de transporte, con el fin de buscar los sectores conflictivos que deberían tener en vista y consideración del personal encargado de manejar el transito, además de la asignación de recursos hacia estas calles según la importancia que podrían a llegar a tener aquellas.

Finalmente la obtención de una nueva red y las mejoras que se utilizaron dan como guía en base a todos las consideraciones tomadas en el transcurso de la memoria, cuales podrían ser los cambios de dirección o dar paso a ciertas calles rompiendo las reglas comunes con el fin de obtener una salida positiva para el caso de catástrofes, de esta formas las personas puedan tener la confianza de romper la regla si es que se presenta una evacuación, al igual de saber si es que el tomar cierta ruta terminara siendo de ayuda para el flujo del trafico en general, al igual de poseer una guía clara para las entidades encargadas de manejar el trafico al evacuar, siendo esto un punto importante para el desarrollo inteligente de una ciudad y que sin duda con el paso del tiempo tomara cada vez mas relevancia.

REFERENCIAS BIBLIOGRÁFICAS

[Castro *et al.*, 2019] Castro, S., Poulos, A., Herrera, J. C., y de la Llera, J. C. (2019). Modeling the impact of earthquake-induced debris on tsunami evacuation times of coastal cities. *Earthquake Spectra*, 35(1):137–158.

[Dong *et al.*, 2019] Dong, H., Zhou, M., Wang, Q., Yang, X., y Wang, F.-Y. (2019). State-of-the-art pedestrian and evacuation dynamics. *IEEE transactions on intelligent transportation systems*, 21(5):1849–1866.

[Dyn, 2001] Dyn, C. F. (2001). M. abramowitz, ia stegun, eds. handbook of mathematical functions with for-mulas, graphs, and mathematical tables, dover (1972). rk ahuja, tl magnanti, jb orlin, tl magnanti, network flows: Theory, al-gorithms, and applications, prentice hall (1993). r. albert, al barabási,"statistical mechanics of complex networks". rev. mod. *Phys. Rev. E*, 64:036106.

[Goodchild, 2007] Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221.

[Hamacher y Tjandra, 2001] Hamacher, H. W. y Tjandra, S. A. (2001). Mathematical modelling of evacuation problems: A state of art.

[Herschelman *et al.*, 2021] Herschelman, R., Qutbuddin, A., y Yang, K. (2021). Conflict-free evacuation route planning. *GeoInformatica*, 25(4):655–678.

[Herschelman y Yang, 2019] Herschelman, R. y Yang, K. (2019). Conflict-free evacuation route planner. En *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 480–483.

[Kim *et al.*, 2007] Kim, S., George, B., y Shekhar, S. (2007). Evacuation route planning: scalable heuristics. En *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pp. 1–8.

[Kovacs, 2007] Kovacs, G. & Spens, K. . (2007). Humanitarian logistics in disaster relief operations. international journal of physical distribution & logistics management. pp. 99–114.

[Liu *et al.*, 2022] Liu, L., Jin, H., Liu, Y., y Zhang, X. (2022). Intelligent evacuation route planning algorithm based on maximum flow. *International Journal of Environmental Research and Public Health*, 19(13):7865.

[Lu *et al.*, 2005] Lu, Q., George, B., y Shekhar, S. (2005). Capacity constrained routing algorithms for evacuation planning: A summary of results. En *International symposium on spatial and temporal databases*, pp. 291–307. Springer.

[Lu *et al.*, 2007] Lu, Q., George, B., y Shekhar, S. (2007). Evacuation route planning: a case study in semantic computing. *International Journal of Semantic Computing*, 1(02):249–303.

[Mordvintsev *et al.*, 2014] Mordvintsev, A., Krzhizhanovskaya, V., Lees, M., y Sloot, P. (2014). Simulation of city evacuation coupled to flood dynamics. En *Pedestrian and Evacuation Dynamics 2012*, pp. 485–499. Springer.

[municipalidad de Valparaíso, 2018] municipalidad de Valparaíso, I. (2018). Plan maestro para la gestión del riesgo deincendios valparaíso.

[Onemi, a] Onemi. Storymap.

[Onemi, b] Onemi. Visoronemo.

[Pan *et al.*, 2007] Pan, X., Han, C. S., Dauber, K., y Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2):113–132.

[Veeraswamy *et al.*, 2018] Veeraswamy, A., Galea, E. R., Filippidis, L., Lawrence, P. J., Haasanen, S., Gazzard, R. J., y Smith, T. E. (2018). The simulation of urban-scale evacuation scenarios with application to the swinley forest fire. *Safety Science*, 102:178–193.

[Yang *et al.*, 2012] Yang, K., Gunturi, V., y Shekhar, S. (2012). A dartboard network cut based approach to evacuation route planning: A summary of results. En *International Conference on Geographic Information Science*, pp. 325–339. Springer.

[Zheng *et al.*, 2009] Zheng, X., Zhong, T., y Liu, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445.