# Pontificia Universidad Católica del Perú

## Facultad de Ciencias e Ingeniería


# ELABORACIÓN DE UN SIMULADOR PARA EL SISTEMA DE ATENCIÓN A EMERGENCIAS BRINDADA POR EL CUERPO GENERAL DE BOMBEROS VOLUNTARIOS DEL PERÚ APLICADO EN LIMA Y CALLAO

Tesis para optar el Título de **Ingeniero Industrial**, que presentan los bachilleres:


**Hector Mattos Galarza**

**Mónica Milagros Huayta Durand**

**ASESOR: Jonatán Edward Rojas Polo**



Lima, noviembre de 2018

Esta tesis es solo una pequeña parte del camino de aprendizaje que hemos decido tomar. Sin embargo, este hito no hubiese sido posible sin la ayuda de todas las personas que nos apoyaron y ayudaron a seguir adelante.

A mis padres, que me permitieron tomar las decisiones más adecuadas guiadas por su amor y experiencia. A mi hermana y mi ahijado, quienes siempre me muestran el lado más alegre de la vida y a mis profesores, quienes gracias a su vocación me enseñaron el valor de la educación.

Hector Mattos

A mi familia, por el apoyo incondicional que siempre me han brindado, a mis mejores amigos, por ser mis ángeles en la universidad y a mis profesores, por sus buenos consejos de vida.

Mónica Huayta

Un agradecimiento especial a Jonatan Rojas, nuestro asesor, quien nos acompañó a lo largo de este camino.

## RESUMEN

Esta tesis tiene por finalidad la elaboración de un simulador para el sistema de atención a emergencias brindaba por el Cuerpo General de Bomberos Voluntarios del Perú, aplicado en Lima y Callao, donde se buscará minimizar el tiempo total de atención, redistribuyendo los recursos y tomando en cuenta diversos factores como el tráfico vehicular, la infraestructura vial, la disposición de las estaciones de bomberos, entre otros.

Actualmente el tiempo de atención, el cual contabiliza el tiempo total entre el primer contacto telefónico con la estación y la primera llegada al lugar de la emergencia, se encuentra en 13 minutos. Este tiempo es mayor al que establece como estándar la Asociación Nacional de Protección contra el fuego, el cual fluctúa de 4 a 9 minutos como máximo. Se identificó como una de las causas principales para la problemática la deficiente distribución de las estaciones y sus recursos.

El modelo desarrollado permite la modificación de diversas variables claves en la sensibilidad del tiempo de atención como el número de vehículos disponibles por estación, el número de estaciones y su distribución, entre otras. Las ventajas del uso de este modelo sobre otros, es que los resultados se aproximan en mayor medida a la realidad, debido a que se utiliza información muy precisa del mapa de Lima y Callao, lo que permite conocer el impacto de las posiciones de las estaciones y emergencia en el tiempo de atención. De la misma manera, se busca mayor precisión con la inclusión del tráfico de Lima de acuerdo con el día y hora para el cálculo del tiempo de traslado de los vehículos.

En la memoria descriptiva se plantea un caso de utilización del modelo para seleccionar la posición de una nueva estación a partir de cuatro propuestas. En esta aplicación se explican los pasos a seguir para usar el modelo y se observan las diferencias significativas en términos del KPI elegido entre las propuestas que conllevan a la selección de la más conveniente; sin embargo, este ejemplo se puede aplicar para otros escenarios y variaciones de los inputs e indicadores de performance.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## TEMA DE TESIS

| | | |
|---|---|---|
| PARA OPTAR | : | Título de Ingeniero Industrial |
| ALUMNO | : | **HECTOR MATTOS GALARZA**<br>**MÓNICA MILAGROS HUAYTA DURAND** |
| CÓDIGO | : | 2010.0451.12<br>2010.0913.12 |
| PROPUESTO POR | : | Ing. Jonatán E. Rojas Polo |
| ASESOR | : | Ing. Jonatán E. Rojas Polo |
| TEMA | : | ELABORACIÓN DE UN SIMULADOR PARA EL SISTEMA DE ATENCIÓN A EMERGENCIAS BRINDADA POR EL CUERPO GENERAL DE BOMBEROS VOLUNTARIOS DEL PERÚ APLICADO EN LIMA Y CALLAO |
| Nº TEMA | : | # 1395 |
| FECHA | : | San Miguel, 07 de julio de 2017 |

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Facultad de Ciencias e Ingeniería
12 JUL. 2017
MIGUEL MEJIA PUENTE
DECANO

**JUSTIFICACIÓN:**

A medida que las ciudades van creciendo, se ven más afectadas por diversos problemas como la inseguridad ciudadana, los accidentes en los hogares o en medios de transporte, entre otros; ya que estas emergencias son más difíciles de ser atendidas en el momento adecuado. Cada uno de estos hechos demanda una atención especializada de acuerdo al tipo de emergencia ocurrida; por ejemplo, en un accidente de tránsito se puede requerir desde la presencia de un efectivo policial hasta la de ambulancias, serenazgos y camiones de bomberos, y puede variar dependiendo del grado de la emergencia ocurrida y los protocolos establecidos por la autoridad local.

En el caso a analizar, el Cuerpo General de Bomberos Voluntarios del Perú (CGBVP) brinda la atención a emergencias dentro de Lima y Callao, siendo la entidad más importante para poder auxiliar esta necesidad. Debido a la naturaleza de estos eventos, la escasez de recursos y tomando en cuenta la teoría de "la hora de oro" [1], el tiempo de servicio debe ser el mínimo posible.
Según las estadísticas brindadas por el CGBVP, el tiempo promedio de servicio o de llegada a un accidente en Lima y Callao se encuentra en 12 minutos, no obstante, según los estándares mundiales[2], se plantea que el tiempo debe ser de 4 a 9 minutos como máximo,

[1] Cowley RA. A total emergency medical system for the state of Maryland. Md State Med J 1975;45:37-45.

[2] NFPA 1720, standard for the organization and deployment of fire suppression operations, emergency medical operations, and special operations to the public by volunteer fire departments. (2014). Quincy, MA: National Fire Protection Association.



Av. Universitaria N° 1801, San Miguel
T: (511) 626 2000
www.pucp.edu.pe

con lo que se concluye que el tiempo actual promedio no es el deseado. Asimismo, al comparar esta cifra con la de otros países como Estados Unidos (10 minutos) y Canadá (7 minutos) queda expuesto que el tiempo de respuesta en las ciudades de Lima y Callao está muy elevado. Como principal causa de estos altos tiempos se tiene a la deficiente asignación de recursos.
A partir de la investigación preliminar, se ha determinado que una de las principales causas de la problemática es la deficiente asignación de los limitados recursos. Es por ello que una mejora en la asignación de recursos, mediante el uso de métodos matemáticos y heurísticos, tendrá un impacto positivo sobre la atención brindada por el CGBVP. Debido a la complejidad del sistema, la utilización de un simulador, el cual reflejará el impacto de la variación de los recursos sobre el sistema y el tiempo de servicio, favorece a la mitigación del problema de una manera eficiente y certera.

**OBJETIVO GENERAL:**

Proponer un método de simulación para sistemas de atención de emergencia enfocado al Cuerpo General de Bomberos Voluntarios del Perú que permita encontrar, analizar y evaluar nuevas propuestas de distribución de recursos y así mejorar el nivel de atención actual.

**OBJETIVOS ESPECÍFICOS:**

- Definir los fundamentos teóricos relacionados a la simulación de eventos discretos, estadística y herramientas de optimización que se utilizarán para el desarrollo del algoritmo.
- Determinar las variables influyentes en el sistema de atención a emergencias.
- Determinar los criterios de validez de los datos requeridos por el simulador.
- Definir los criterios de depuración para los datos requeridos.
- Proponer un método eficiente en términos de procesamiento para llevar a cabo la simulación.
- Realizar la validación del sistema.
- Plantear propuestas de mejora del nivel de atención en el caso analizado para probar la utilidad del simulador.

**PUNTOS A TRATAR:**

a. **Marco Teórico. [MMHD-HMG]**
Se explicarán y detallarán los conceptos básicos de teoría de grafos para poder interpretar los términos utilizados en la tesis y los algoritmos presentados. Por otro lado, se presentarán los conceptos de los algoritmos empleados para la simulación como el de Dijkstra, Quicksort, Breadth First Search, entre otros.

Por último, se repasarán las nociones generales de la simulación de eventos discretos y de bondad de ajuste con los cuales se desarrollará y validará el simulador presentado.



Av. Universitaria N° 1801, San Miguel
T: (511) 626 2000
www.pucp.edu.pe

**b. Descripción y definición del proyecto. [MMHD-HMG]**
Se presentarán casos de estudios que abordan temas de modelos de optimización, de los cuales se recaben conceptos que se puedan utilizar en la tesis.

**c. Diagnóstico de la situación actual. [MMHD-HMG]**
Se pone en contexto los antecedentes de las emergencias de Lima y Callao, se analizar las causas de la problemática, se proporcionan los protocolos utilizados por el CGBVP y los tiempos actuales de servicio con los que se cuenta. Asimismo, estos se contrastar con los tiempos de los estándares mundiales y de otros países.

**d. Aplicación de la metodología propuesta. [MMHD-HMG]**
Se explica detalladamente los pasos seguidos para la depuración de las bases de datos utilizados en la tesis, el tratamiento, el pre-procesamiento y la pre-simulación respectiva. Además, se muestra la metodología utilizada para simulación y la validación del sistema.

**e. Aplicación del modelo y propuesta de mejora. [MMHD-HMG]**
Con el simulador validado, se realizan variaciones de los recursos para medir el impacto sobre el sistema y el tiempo de servicio, con lo que posteriormente se selecciona la propuesta que represente el mejor resultado según los indicadores empleados en el desarrollo de la tesis.

**f. Conclusiones y recomendaciones.**

Máximo: 100 páginas

---
**ASESOR**

**[MMHD] MÓNICA HUAYTA DURAND**
**[HMG] HECTOR MATTOS GALARZA**

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

# CAPÍTULO 1. MARCO TEÓRICO

En este capítulo se detallará los principales conceptos relacionados con el tema de tesis a desarrollar.

## 1.1. Teoría de grafos

A continuación, se exponen algunos conceptos de la teoría de grafos.

### 1.1.1. Grafo

Un grafo es un par $G = (V, E)$ de conjuntos finitos, compuesto por $V \neq \emptyset$, cuyos elementos son llamados nodos o vértices, y por *E*, que contiene a $e = \{a, b\}$, el cual es un eje que inicia en el vértice $a$ y termina en el vértice $b$, pertenecientes a V. También se denota a los nodos $a$ y $b$ como incidentes con el eje $e$ y como nodos vecinos o adyacentes entre ellos mismos, y se representa como $e = ab$.

Según Jungnickel (2008), de un grafo $G$ se pueden obtener 3 tipos de subgrafos:

- Subgrafo

Sea un grafo $G = (V, E)$ y $V'$es un subconjunto de $V$. Por $E|V'$denotamos el conjunto de todos los ejes $e \in E$ que tienen ambos vértices en $V'$. En la Figura 1.B se muestra un subgrafo resultante de la Figura 1.A.

- Subgrafo Inducido

Un grafo $(V', E|V')$ es llamado subgrafo inducido en $V'$ y es representado por $G|V'$. Cada grafo de la forma $(V', E')$ donde $V' \subset V$ y $E' \subset E|V'$ es llamado subgrafo de $G$. Es decir, tendrá los mismos vértices del subgrafo de la Figura 1.B y tendrá todas las aristas que puedan conectarse a los vértices que quedan en el subgrafo inducido, que están incluidas en el conjunto de aristas originales. En la Figura 1.C se observa un ejemplo de subgrafo inducido.

- Subgrafo abarcado

Un subgrafo con $V' = V$ es llamado un subgrafo abarcado, es decir, cuenta con los mismos vértices del grafo inicial de la Figura 1.A. En la figura 1.D se representa al subgrafo abarcado.

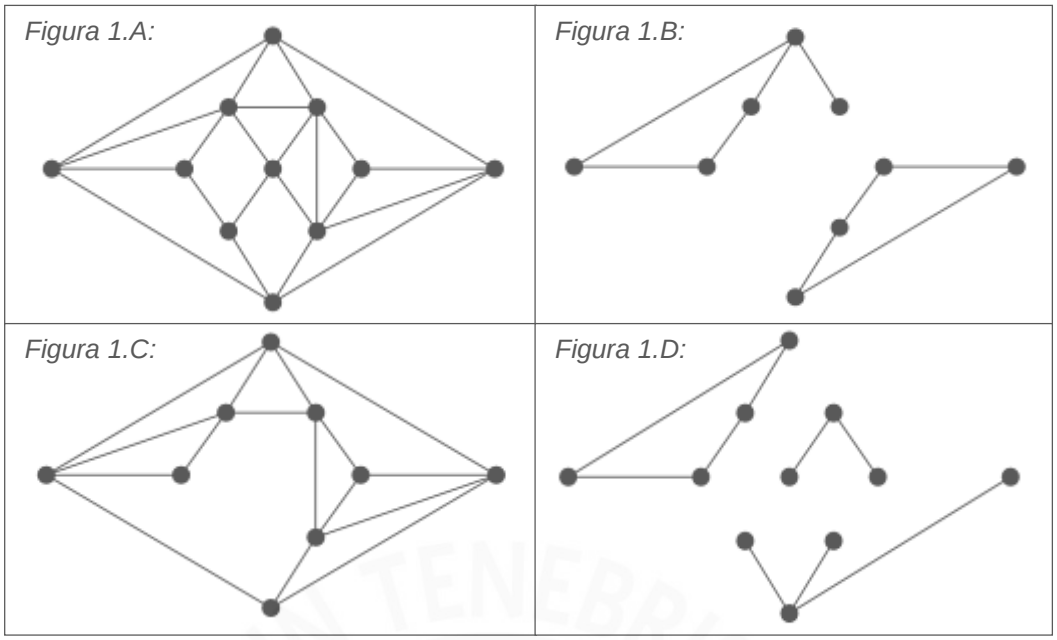



*Figura 1: Tipos de grafos*
*Fuente: Jungnickel (2008)*

### 1.1.2. Grafos no direccionados

Un grafo no direccionado o también llamado grafo no dirigido es un grafo cuyas aristas no tienen orientación alguna. La Figura 1.A ilustra un grafo no direccionado.

### 1.1.3. Grafos direccionados

Un grafo direccionado o también llamado dígrafo es un par $G = (V, E)$ que está compuesto por un conjunto finito de vértices $V$ y arcos $E$ que están ordenados en pares $(a, b)$ donde a $\neq b$ son vértices de $V$. En este caso, $e = (a, b)$ donde $a$ es llamado vértice de inicio y $b$ como vértice final. Se dice que $a$ y $b$ son incidentes con $e$ y se les llama a estos dos arcos $ab$ y $ba$ como antiparalelos.

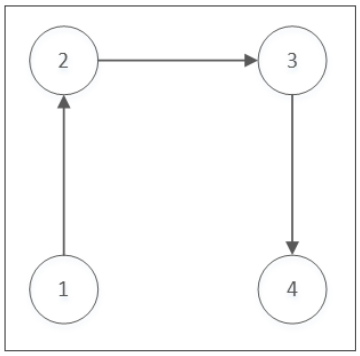



*Figura 2: Grafo direccionado*
*Elaboración propia*

### 1.1.4. Grafos conexos

Se dice que $G$ es un grafo conexo si para todo vértice $a$ y $b$ que pertenecen a $G$ existe por lo menos una ruta de $a$ hacia $b$. En este sentido un grafo conexo con $n$ vértices tiene al menos $n - 1$ arcos. La Figura 2 muestra un grafo conexo.

### 1.1.5. Cadenas y trayectorias

Una cadena es una secuencia de arcos, tal que cada arco tiene un vértice en común con un arco previo. Una trayectoria es una cadena de arcos, de tal manera que el nodo terminal del cada arco es igual al nodo inicial del arco siguiente.

En la Figura 3 se observa el ejemplo de una cadena y trayectoria $(1-2)-(2-4)-(4-6)$, la cual representa una manera de viajar desde el nodo INICIO 1 hasta el nodo FIN 6.

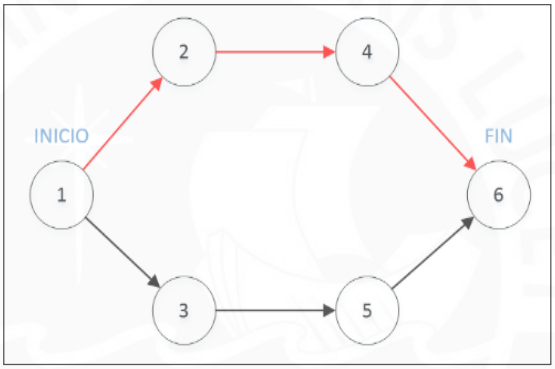



*Figura 3: Cadena y trayectoria*
*Elaboración propia*

## 1.2. Lista de adyacencia

La representación de un grafo $G = (V, E)$ como lista de adyacencia consiste en un arreglo $Adj$ de $|V|$ listas, una para cada vértice $V$. De esta manera, para cada vértice $u \in V$ la lista $Adj[u]$ contiene cada uno de los vértices v para los cuales existe una arista $(u, v) \in E$. En la Figura 4 se representa el grafo (a) y la lista de adyacencia (b).

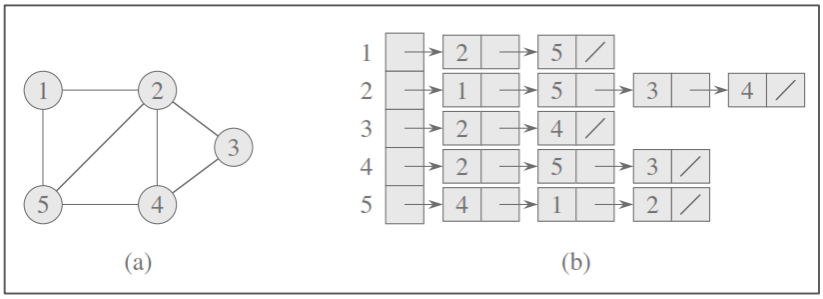



*Figura 4: Lista de adyacencia*
*Fuente: Cormen (2009)*

## 1.3. Búsqueda en anchura

El algoritmo de búsqueda en anchura o breadth first search (BFS) es uno de los algoritmos más simples de la búsqueda de grafos. En este sentido, sea $G = (V, E)$ un grafo o dígrafo definido por una lista de adyacencia, y el nodo origen $s$, el algoritmo busca de manera sistemática lo arcos de $G$ para descubrir cada nodo que es alcanzable desde $s$. Este algoritmo obtiene su nombre debido a que los nodos se van agregando a lo ancho de toda la frontera en cada iteración. A continuación, se presenta el pseudocódigo del algoritmo.

```
Algoritmo de búsqueda en anchura

Q ← ∅; d(s) ← 0;
Anexar s a Q;
Mientras Q ≠ ∅ Hacer;
Remover el primer vértice v de Q;
    Para Cada  w ∈ A_v  Hacer
          Si  d(w)  es indefinido Entonces
          Anexar  w a Q;
          FinSi
    FinPara
FinMientras
```

La importancia del uso del algoritmo BFS dentro del tema de estudio es la de encontrar los nodos que pertenecen al árbol principal, los cuales constituyen el mapa de Lima.

## 1.4. Algoritmo de Dijkstra

Este algoritmo fue presentado por Dijkstra, el cual busca hallar la ruta más corta en un grafo direccionado $G = (V, E)$ donde se cumpla que los pesos de todos los arcos sean positivos $w = (u, v) \geq 0$. A continuación se muestra el funcionamiento del algoritmo.

```
Algoritmo de Dijkstra

Función Dijkstra (G, S)
Crear conjunto de vértices Q
Para v ∈ G Hacer
dist[v] ← ∞
prev[v] ← indefinido
Agregar v a Q
dist [S] ← 0
FinPara
Mientras Q ≠ ∅ Hacer
u ← vértice en Q con dist[u] mínima
Quitar u de Q
    Para cada vecino v de u:
    alt ← dist[u] + longitud(u, v)
    Si alt < dist[v]:
```

```
        dist[v] ← alt
        prev[v] ← u
    FinPara
    FinMientras
```

## 1.5. Distancia más corta en espacios elípticos

La distancia más corta entre dos puntos sobre la superficie de una esfera es determinada por el arco de circunferencia sección de un círculo mayor de la esfera. Para calcular esta distancia se utiliza la ley de Haversine (Semiverseno), donde $hav(x)$ es el Haversiano de $x$.

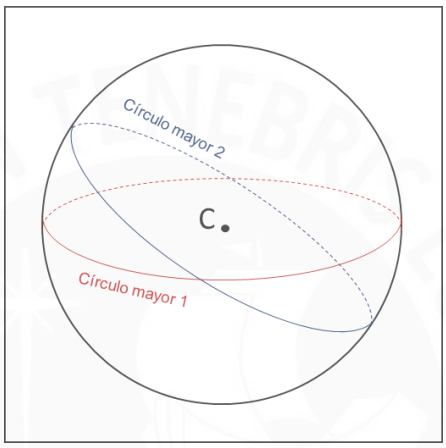



*Figura 5: Representación de dos círculos mayores*
*Elaboración propia*

Dado un triángulo definido sobre la superficie de una esfera unitaria por tres puntos y sus distancias más cortas, como se muestra en la Figura 6, se calcula el Semiverserno de “C” de la siguiente manera:

$$hav(c) = hav(a - b) + \sin(a)\sin(b)\,hav(C)$$

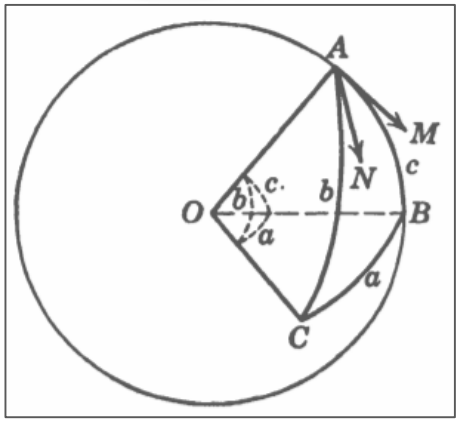



*Figura 6: Ilustración de un triángulo esférico*
*Fuente: Lyman, Willis y James (1940)*

## 1.6. Algoritmo de ordenamiento Quicksort

El algoritmo de ordenamiento Quicksort es una estrategia de divide y vencerás (Sedgewick, 1998), usada muy comúnmente en informática, funciona dividiendo un archivo en dos partes, y ordenando independientemente cada una de ellas. Además, su implementación es sencilla, proporciona buenos resultados y consume menos recursos que cualquier otro método de ordenación.

Este algoritmo de ordenación rápida tiene como ventaja la de trabajar in situ, donde solo necesita el orden de $NlogN$ operaciones en promedio para poder ordenar $N$ elementos y tiene un bucle interno corto.

La estrategia de Quicksort funciona de la siguiente manera:

> Dado un vector de elementos desordenados, podemos elegir un elemento (que llamaremos pivote) y organizar el resto de ellos de tal manera que, a la izquierda de éste se encuentren todos los no más grandes que él, y que a su derecha todos los que no son menores que él, con lo que el pivote ocupará el lugar correcto dentro de la futura secuencia ordenada. A cada uno de los dos subconjuntos podemos aplicar la misma idea y obtener dos elementos, uno por cada subconjunto, que partirá a cada uno de ellos en dos subconjuntos más y que estarán correctamente colocados en el futuro conjunto ordenado. En cada partición se van ordenando los elementos, con lo que al llegar a particiones de un elemento conoceremos el lugar correcto de todos los elementos del conjunto. Si el algoritmo es construido de tal forma que las llamadas no dependen unas de otras, como es el caso, al llegar a estas particiones de un solo elemento, el conjunto estará ordenado. (Martínez y Quetglás, 2003, p.238)

A continuación, se muestra el pseudo-código del algoritmo de ordenamiento.

```
Algoritmo de Quicksort

Función [ salida ] = quicksort( lista )
fijados = zeros(1,size(lista,2));
salida=lista;
Mientras sum(fijados)<size(fijados,2)
i=find(~fijados,1);
n=i;
Mientras n+1<=size(fijados,2) && fijados(1,n+1)==0
n=n+1;
Fin
izq=0;
der=0;
analizando=lista(1,n);
```

```
Para m=i:n-1
Si lista(1,m)<analizando
Salida(1,i+izq)=lista(1,m);
izq=izq+1;
else
salida(1,n-der)=lista(1,m);
der=der+1;
Fin
Fin
salida(1,i+izq)=analizando;
fijados(1,i+izq)=1;
lista=salida;
Fin
Fin
```

## 1.7. Simulación de eventos discretos

La simulación de eventos discretos consiste en el modelamiento de un sistema a medida que evoluciona en el tiempo en donde los estados de las variables cambian instantáneamente en puntos de tiempo separados (Law, 1991). Estos puntos en el tiempo son los únicos en los cuales un evento (ocurrencia que puede hacer cambiar el estado del sistema) puede ocurrir. Existen dos tipos de sistemas, los cuales se distinguen de acuerdo con la data de salida de la simulación: sistemas terminales y sistemas no terminales.

### 1.7.1. Sistemas terminales

Un sistema terminal es aquel que comienza en el tiempo $0$ y se ejecuta durante un tiempo $T_E$, donde $E$ es un evento específico o una serie de eventos que detienen la simulación (Banks, 2010).

### 1.7.2. Sistemas no terminales

Un sistema no terminal es aquel que se ejecuta continuamente o por un largo periodo de tiempo. La simulación de este tipo de sistemas comienza en el tiempo $0$ bajo condiciones iniciales definidas y se ejecuta por un periodo de tiempo $T_E$ definido de acuerdo con el sistema, donde usualmente se desea analizar desde el estado estable (Banks, 2010).

### 1.7.3. Características del estado estable

Las características del estado estable son aquellas que se obtienen al ejecutar la simulación de un sistema no terminal por un largo periodo de tiempo. Suponiendo que una simulación produce las observaciones $Y_i$ que, generalmente, corresponden

a la muestra de una serie de tiempo autocorrelacionada. La medida de desempeño $\theta$ de estado estable es definida como:

$$\theta = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} Y_i$$

El valor de $\theta$ es independiente a las condiciones iniciales del sistema (Banks, 2010).

### 1.7.4. Método del batch mean

En una simulación no terminal, generar una sola réplica por un tiempo largo ocasiona problemas al calcular el error estándar de la media de la muestra, los datos presentan elevada correlación y el estimador de la media o varianza puede estar sesgado. El método del batch mean trata de solucionar este problema, dividiendo la data de salida de una réplica en $k$ lotes grandes y tomando el promedio de estos lotes como independientes, todo esto luego de haber eliminado los datos del periodo de calentamiento (Banks, 2010), para así poder estimar el promedio del estado estable de $\{X_i\}$. Según Schmeiser (1982), no hay razón para considerar un número de lotes mayor a 30.

Al obtener la data de salida de una simulación larga $X_1, \dots, X_n$, se obtiene $E(X_i) = \mu$, $Var(X_i) = \sigma^2$ y la $Cov(X_i, X_j)$, que dependerá del "lag" o retraso $|j - i|$ (Alexopolus y Seila, 1996). Al separar la data en $k$ lotes de $b$ observaciones. El $i - ésimo$ lote contendrá las siguientes observaciones:

$$X_{(i-1)b+1}, X_{(i-1)b+2}, \dots, X_{ib}$$

Para $i = 1,2, \dots, k$. El $i - ésimo$ promedio del lote está dado por:

$$Y_i(b) = \frac{1}{b}\sum_{j=1}^{b} X_{(i-1)b+j}$$

El promedio de los lotes estará dado por:

$$\bar{Y}_k = \bar{X}_n = \frac{1}{k}\sum_{i=1}^{k} Y_i(b)$$

La varianza estará dada por:

$$\hat{V}_B(n,k) = \frac{1}{k-1}\sum_{i=1}^{k} (Y_i(b) - \bar{Y}_k)^2$$

Y el intervalo de confianza $1 - \alpha$ para la media $\mu$ es:

$$\bar{Y}_k \pm t_{k-1,1-\frac{\alpha}{2}}\sqrt{\frac{\hat{V}_B(n,k)}{k}}$$

Donde $t_{k-1,1-\frac{\alpha}{2}}$ sigue una distribución t-student con $k-1$ grados de libertad y nivel de significancia $\frac{\alpha}{2}$.

### 1.7.5. Correlogramas y autocorrelación

Un correlograma es la herramienta gráfica donde se representa la correlación de un conjunto de datos, para valores en distintos intervalos de tiempo determinados por un "lag" o retraso (Gujarati, 2010). Por ejemplo, para hallar la correlación con el lag-1 se utiliza lo siguiente:

$$\hat{\rho}_1 = \frac{\sum_{j=1}^{k-1}(\bar{Y}_j - \bar{Y})(\bar{Y}_{j+1} - \bar{Y})}{\sum_{j=1}^{k}(\bar{Y}_j - \bar{Y})^2}$$

Donde $k$ es el número de batches, $\bar{Y}$ es el promedio de los batches y $\bar{Y}_j$ es el valor del $j-ésimo$ batch.

Según Banks (2010), se propone lo siguiente:

- Si $\hat{\rho}_1 \leq 0.2$, se realiza una segunda agrupación en $30 \leq k \leq 40$ lotes, a partir de los lotes iniciales. Luego, se hallan intervalos de confianza con los nuevos lotes y $k-1$ grados de libertad. Adicionalmente, se puede realizar una nueva prueba de independencia de los lotes utilizando la siguiente fórmula:

$$C = \sqrt{\frac{k^2-1}{k-2}}\left(\hat{\rho}_1 + \frac{(\bar{Y}_1 - \bar{Y})^2 + (\bar{Y}_k - \bar{Y})^2}{2\sum_{j=1}^{k}(\bar{Y}_j - \bar{Y})^2}\right)$$

- Si $c < Z_\beta$, se acepta la independencia de las medias de los lotes, donde $Z_\beta$ es el valor de una variable normal estándar con nivel de confianza $\beta$.
- Si $\hat{\rho}_1 > 0.2$, se debe extender el tiempo se simulación en un 50% a 100% y volver a realizar el análisis. En caso no sea posible prolongar el tiempo de simulación, se reagrupan los lotes en $k = 10$ nuevos lotes y se procede a hallar los intervalos de confianza con $k-1$ grados de libertad.

### 1.7.6. Intervalos de confianza

El intervalo de confianza es el rango, con amplitud relativamente pequeña, compuesto por un límite superior y un límite inferior, dentro del cual se encuentra el parámetro objetivo $\theta$. La probabilidad de que el parámetro se encuentre dentro de este intervalo de denomina coeficiente de confianza (Mendehall, 2009). Para el

desarrollo de la tesis se utilizará el intervalo de confianza para una muestra grande, donde los estimadores puntuales tienen distribuciones muestrales aproximadamente normales, entonces se tiene que:

$$Z = \frac{\hat{\theta} - \theta}{\sigma_\theta}$$

$$Z \sim N(0,1)$$

Donde $\hat{\theta}$ es el estadístico distribuido normalmente con media $\theta$ y error estándar $\sigma_\theta$. El intervalo de confianza para $\theta$ con un coeficiente de confianza igual a $(1-\alpha)$ está dado por:

$$I.C. = [\hat{\theta} - z_{\alpha/2}\sigma_\theta, \hat{\theta} + z_{\alpha/2}\sigma_\theta]$$

Los intervalos de confianza servirán para la comparación y validación de las propuestas finales de la tesis.

## 1.8. Bondad de ajuste

Las pruebas de bondad de ajuste se usan para probar estadísticamente que una distribución de frecuencias observadas es compatible o se ajuste con alguna distribución teórica conocida como la uniforme, multinomial, binomial, poisson, normal, etc. Entre las principales pruebas de bondad de ajuste se tiene a la Chi-Cuadrado y Kolmogorov - Smirnov.

### 1.8.1. Prueba de bondad Kolmogorov - Smirnov

La prueba de bondad de ajuste Kolmogorov - Smirnov es una prueba no paramétrica utilizada para determinar si dos muestras provienen de una población con distribución específica. Las hipótesis que se plantean son:

$$H_0: La\ data\ sigue\ la\ distribución\ especificada$$
$$H_1: La\ data\ no\ sigue\ la\ distribución\ especificada$$

El estadístico de prueba se define por:

$$D = max_{1 \le i \le N}\left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i)\right)$$

Donde $F$ es la distribución acumulada teórica a ser probada, la cual debe ser continua y todos sus parámetros deben estar completamente definidos. La hipótesis nula se rechaza si el estadístico de prueba $D$ es mayor que el valor crítico obtenido por tabla.

Las principales limitantes de este tipo de prueba de bondad de ajuste son:

- Solo aplica en distribuciones continuas.
- Es más sensible en el centro de la distribución que en las colas.
- La distribución debe tener todos los parámetros definidos. Si estos son estimados desde la data, la región crítica no es válida. Los parámetros deben ser definidos por simulación.

### 1.8.2. Prueba de bondad Chi - Cuadrado

La prueba de bondad de ajuste Chi – Cuadrado es una prueba utilizada para determinar si dos muestras provienen de una población con distribución específica. Este tipo de prueba puede ser aplicada cualquier distribución univariada de la que se puede obtener su distribución acumulada. La principal ventaja de este tipo de prueba frente a la Kolmogorov - Smirnov es que puede ser aplicada a distribuciones discretas como la binomial y la poisson. Para aplicar esta prueba, los datos deben ser agrupados en $k$ intervalos de clase.

Las hipótesis que se plantean son:

$$H_0: La\ data\ sigue\ la\ distribución\ especificada$$
$$H_1: La\ data\ no\ sigue\ la\ distribución\ especificada$$

Es estadístico de prueba se define por:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Donde $O_i$ y $E_i$ son la frecuencia observada y la frecuencia esperada en el intervalo de clase i-ésimo respectivamente. La frecuencia esperada se obtiene con:

$$E_i = N(F(Y_u) - F(Y_l))$$

Donde $F$ es la distribución acumulada teórica a ser probada, $Y_u$ es el límite superior para la clase $i$, $Y_l$ es el límite inferior para la clase $i$, y $N$ es el tamaño de la muestra.

El test estadístico sigue aproximadamente una distribución Chi – Cuadrado con $k - c$ grados de libertad, donde $k$ es el número de intervalos de clase y $c$ es el número estimado de parámetros. La hipótesis nula es rechazada si:

$$\chi^2 > \chi^2_{1-\alpha,k-c}$$

Donde $\chi^2_{1-\alpha,k-c}$ es el valor crítico de Chi – Cuadrado con $k - c$ grados de libertad y nivel de significancia $\alpha$.

## 1.9. Outliers

Es un valor atípico dentro de una observación que es diferente al resto. Al no eliminar este tipo de datos, se generan estadísticas incorrectas. Para el caso de la tesis se utilizará principalmente la Prueba de Tukey, basado en lo siguiente:

Se tiene a Q1 como primer cuartil, Q3 como tercer cuartil y a la diferencia de ambos como rango intercuartil. Se considerará un valor atípico aquel que se encuentre 1.5 veces el rango intercuartil alejados de uno de esos cuartiles.

$$Q_1 = Primer\ cuartil$$
$$Q_3 = Tercer\ cuartil$$

$$IQR = Q_3 - Q_1 = Rango\ intercuartil$$
$$valor\ atípico < Q_1 - 1.5xIQR$$

$$valor\ atípico > Q_3 + 1.5xIQR$$

## 1.10. Estimación de densidad de Kernel

La estimación de densidad de Kernel univariada ($KDE$) es un método no paramétrico para estimar la función de densidad de probabilidad de una variable aleatoria (Silverman, 1986). Este tipo de estimación, utilizado generalmente en minería de datos y econometría, es un problema fundamental de suavizado de datos donde las inferencias sobre la población son basadas en datos finitos.

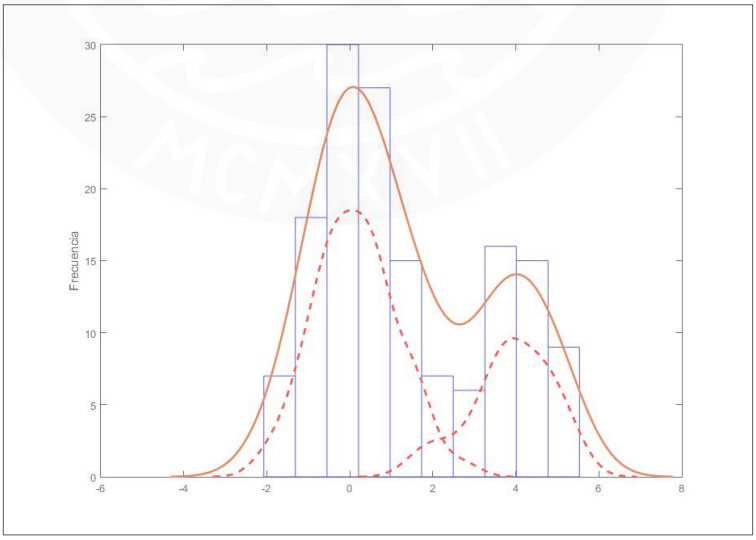



*Figura 7: Representación de las componentes normales de la densidad de Kernel*
*Elaboración propia*

Se tiene $(X_1, X_2, \dots, X_i)$ la cual es una muestra independiente e idénticamente distribuida extraída de una función de densidad $f$ desconocida. Para poder estimar la función se utilizará el siguiente estimador:

$$\hat{f}(x,h) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)$$

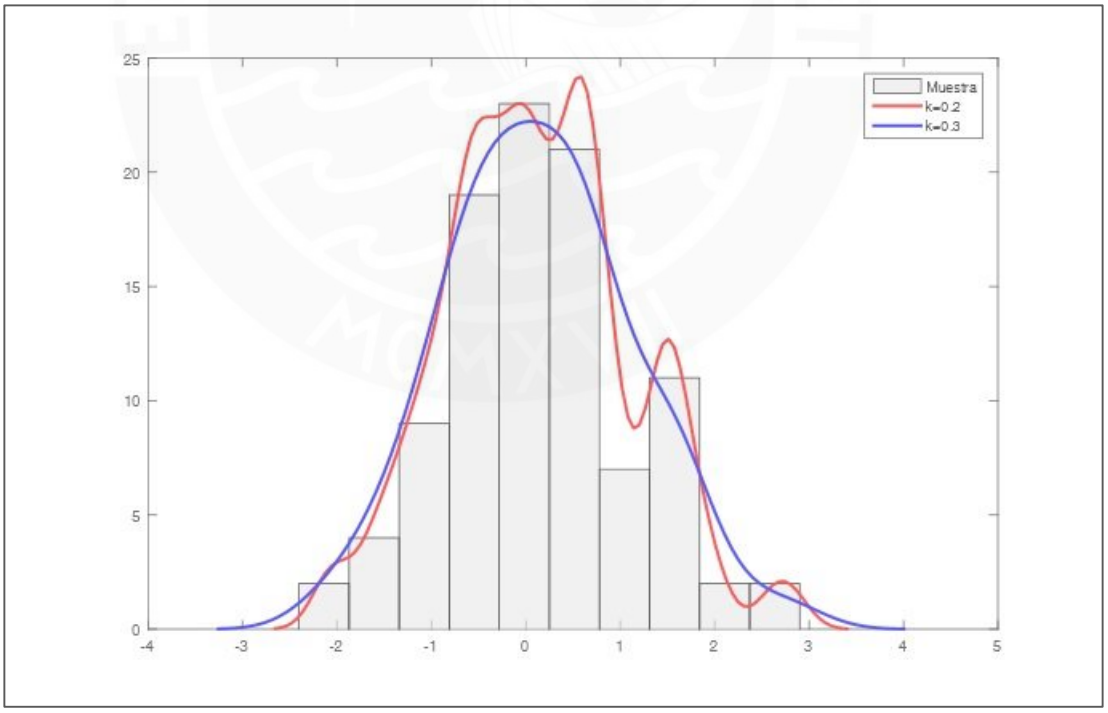
$$K_h(u) = h^{-1}K(h^{-1}u)$$

Donde $K$ es Kernel (una función no negativa que al ser integrada en $\mathbb{R}$ resulta 1 y cuya media es 0) y $h > 0$ es un parámetro suavizado llamado ancho de banda. En el caso del parámetro $h$, se sugiere elegir el más pequeño como los datos lo permitan.

Dado que usualmente Kernel tiene la forma de una distribución normal estándar $N(0,1)$, finalmente se obtiene como estimador a:

$$\hat{f}(x,h) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)}h} \exp(-\frac{(x - X_i)^2}{2h^2})$$

En la Figura 8 se muestra la distribución original de una serie de datos comparado con distribuciones halladas con Kernel y con diferentes anchos de banda.




H

*Figura 8: Ajuste Kernel con diferentes anchos de banda*
*Elaboración propia*

En el caso de la tesis, además del univariado, se utilizará la distribución Kernel multivariado donde el estimador de la función es el siguiente:

$$\hat{f}(x,h) = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{H}|^{-1/2} K\left(\mathbf{H}^{-\frac{1}{2}}(x - X_i)\right) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(x - X_i)$$

Y:

$$K_{\mathbf{H}}(x) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-\frac{1}{2}}x)$$

Donde $\mathbf{H}$ es la matriz del ancho de banda $dxd$, la cual es simétrica no aleatoria y positiva, $d$ es la cantidad de dimensiones y $x = (x_1, x_2, \dots, x_d)^T$ y $X_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$, $i = 1,2, \dots, n$ es una secuencia de d-variables aleatorias independientes e idénticamente distribuidas extraídas de la función de densidad $f$.

# CAPÍTULO 2. CASOS DE ESTUDIO

En el siguiente capítulo se mostrará investigaciones realizadas por diversos autores relacionados al tema principal de la tesis, de esta manera se podrá tener un mejor conocimiento de las herramientas que ya han sido utilizadas.

## 2.1. Caso 1: Reducción de tiempo de respuesta en ambulancias

**Nombre:** "Reducing Ambulance Response Times Using Discrete Event Simulation"
**Autor:** Sean Shao Wei Lam PhD, Zhong Cheng Zhang BEng, Hong Choon Oh PhD, Yih Ying Ng MBBS (S'pore), MRCS A&E (Edinburgh), MPH, MBA, Win Wah MBBS (Yangon), MPH, Marcus Eng Hock Ong MBBS (S'pore)
**Página web:** http://dx.doi.org/10.3109/10903127.2013.836266

El objetivo de este estudio es el desarrollo de un modelo de simulación de eventos discretos para el servicio médico de emergencias de Singapur que permita evaluar distintas alternativas para mejorar los tiempos de respuestas de las ambulancias. El modelo se desarrolló con información recolectada durante 6 meses y su principal objetivo principal es la reducción de los tiempos de respuesta y mantener la utilización de los recursos a niveles razonables.

Se analizaron 3 distintos escenarios en los cuales se realizaron las siguientes modificaciones respectivamente:

- Reposicionamiento y reprogramación de ambulancias
- Agregar ambulancias
- Envío de policías en situaciones concretas previa evaluación en vez de ambulancias

En cuanto a los resultados, la mayor mejora se dio con una estrategia de aumento del número de ambulancias junto con una reprogramación de los horarios de disponibilidad de los vehículos. Sin embargo, el aumento del número de ambulancias también conllevó a la disminución de la utilización de estos, hecho que no es deseable. Asimismo, se calcula que las 10 ambulancias adicionales propuestas incurren en un costo de aproximadamente US$4-6 millones/año para operar.

Como se puede ver en el siguiente gráfico, las distribuciones de los tiempos de respuesta para ambas estrategias no tienen diferencias significativas. Es por esto por lo que la estrategia de reprogramación de horarios se considera superior.

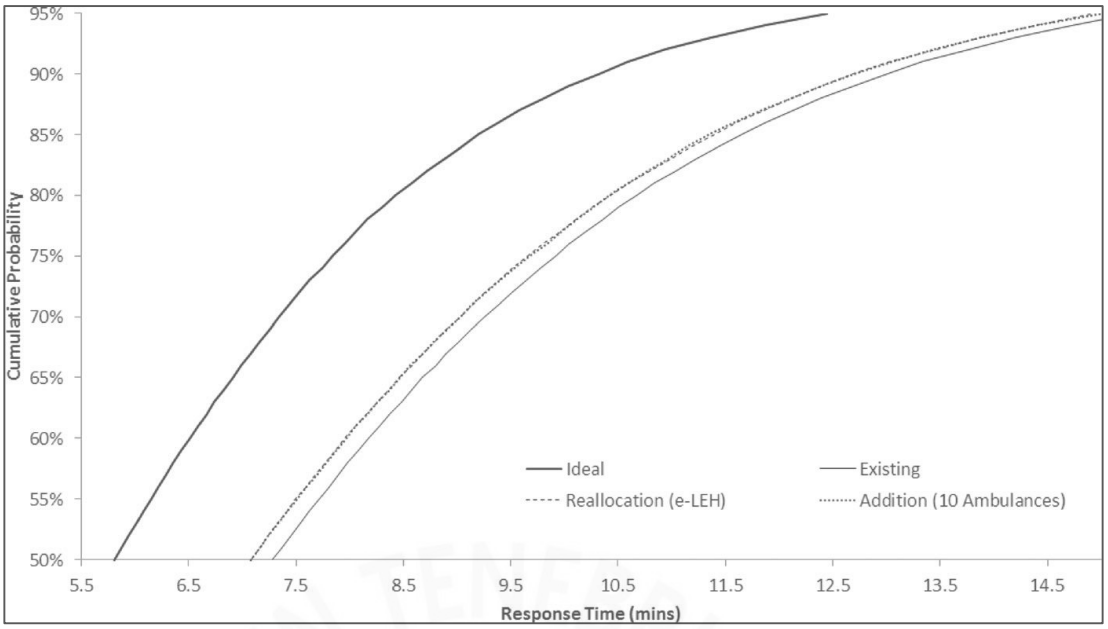



*Figura 9: Tiempo de respuesta vs probabilidad acumulada*

*Fuente: Wei, Zhang, Oh, Wah & Hock (2014)*

En conclusión, se logró demostrar que es posible mejorar el servicio de atención mediante cambios en las políticas de las ambulancias sin necesidad de añadir ambulancias ni reducir los niveles de utilización. Más aún, se evidencia la importancia y validez de la utilización de un modelo de eventos discretos para la facilitación de pruebas de escenarios de manera eficiente, a bajo costo y libre de riesgos.

### 2.2. Caso 2: Patrullaje Policial

**Nombre:** "Police patrol districting method and simulation evaluation using agent-based model & GIS"
**Autor:** Yue Zhang & Donald E. Brown
**Página web:** https://doi.org/10.1186/2190-8532-2-7

El presente caso de estudio se centra en el planteamiento de una estrategia de patrullaje (de policías) eficiente, dado que es un recurso limitado, la cual debe cubrir todas las zonas de Charlottesville al menor costo posible.
Inicialmente, los límites de las áreas de patrullaje eran realizadas a mano, con el conocimiento del departamento de policías, tomando en cuenta que el área a patrullar no debe exceder la capacidad de un solo oficial, las barreras naturales (ríos, puentes, etc.), carga de trabajo y características del área (mayor probabilidad de criminalidad, demografía, etc.).

En el caso de estudio se proponen distintos métodos para el diseño del patrullaje policial y la partición de cada zona a patrullar:

- Unión de partes geográficas pequeñas para formar unidades más grandes llamadas distritos. En este caso, los polígonos formados de cada distrito deben ser compactos (asegura menor tiempo de viaje entre dos puntos tomando en cuenta la distancia Manhattan) y colindantes (ningún individuo se queda fuera de alguna zona).
- Usando teoría de grafos. En este caso, las unidades geográficas se representan por nodos con peso (atributos de la región, población, etc.) y la adyacencia entre regiones como arcos conectados a los nodos. La función objetivo será, por ejemplo, minimizar la diferencia absoluta de la población entre distritos, minimizar la diferencia de delitos predichos entre distritos, minimizar la diferencia de carga de trabajo entre patrullas, etc. Para la optimización se puede hacer uso de métodos heurísticos, sin embargo, solo brindarán óptimos locales que puedes servir para distritos pequeños.

Se hará uso de un simulador para medir los resultados de las propuestas, el cual permite medir el tiempo de respuesta promedio y la carga de trabajo por patrulla. Antes de medir los resultados de las propuestas, se generarán parámetros aleatorios de los límites, y la diferencia entre los parámetros y las medidas de performance serán estudiadas y se obtendrán mejores planes de partición.

En conclusión, con los métodos planteados, no solo se logró reducir significativamente el tiempo de respuesta promedio, sino, también la carga de trabajo por patrullero. Por otro lado, esta simulación no contempla los cambios que pueden suceder dentro de un año de manera efectiva (clima y patrones de tráfico), por lo que se recomienda desarrollar mejores como submuestreo y otros métodos de reducción de datos.

### 2.3. Caso 3: Accidentes de tráfico

**Nombre:** “Análisis espacial de los accidentes de tráfico con víctimas mortales en carretera en España, 2008 – 2011”

**Autor:** Diana Gómez Barroso, Teresa López Cuadro, Alicia Llácer, Rocío Palmera Suarez y Rafael Fernandez Cuenca

**Página web:** https://doi.org/10.1016/j.gaceta.2015.02.009

El presente caso de estudio de centra en hallar las áreas donde existe mayor incidencia de accidentes de tránsito en carretera con víctimas mortales a 24 horas por km$^2$ / año en España peninsular. Este tipo de estudios ya ha sido realizado en otros países donde se analizó la distribución espacial de los accidentes de tránsito utilizando SIG (sistema de información geográfica) y técnicas de análisis espacial para detectar agregaciones de riesgo y excesos de accidentes. Existen diversos métodos de detección de puntos de riesgo, como la densidad de Kernel y K-means, los cuales obtienen resultados muy similares.

Para el caso de estudio se utilizará el método de densidad de Kernel, que en términos matemáticos se expresa como:

$$f(x,y) = \frac{1}{n * h^2} * \sum_{i=1}^{n} k * \frac{d_i}{h}$$

Donde $f(x,y)$ es el valor de densidad en la posición $(x,y)$, $n$ el número de casos, $h$ el ancho de banda, $d_i$ es la distancia geográfica entre el caso $i$ y la ubicación $(x,y)$, y $k$ es la función de densidad.

El autor propone ciertos métodos para tres ámbitos en particular pertinentes al problema propuesto: tratamiento de datos, geo-codificación de los accidentes y análisis espacial. Para el tratamiento de datos, se procedió a la agrupación de las distintas categorías de rutas en solo tres: autovías y autopistas, carreteras convencionales, y otros. Por otro lado, las dimensiones espaciales y temporales del problema, por temas de complejidad, fueron discretizados. En cuanto a la geo-codificación de accidentes, se propuso la utilización de una herramienta en un SIG que permita utilizar los índices de los puntos kilométricos para poder localizar los accidentes de tránsito y obtener las coordenadas “X” e “Y”.

En conclusión, utilizando el método del vecino más cercano, se pudo obtener posibles agrupamientos y así hallar el ancho de banda necesario para calcular la densidad de kernel. Además, con la utilización de estos métodos, se pudo obtener una mejor aproximación de las posiciones de los accidentes de tráfico tomando en cuenta el punto kilométrico, por medio de un mapa, lo cual puede ser una herramienta bastante útil para las autoridades al momento de tomar decisiones.

# CAPÍTULO 3. DESCRIPCIÓN Y DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

En el presente capítulo se describirá los antecedentes de los principales tipos de emergencias en Lima Metropolitana, sus principales causas, la situación actual y la situación futura de la problemática de la tesis.

## 3.1. Antecedentes de las emergencias en el Perú y en Lima

Para el tema de tesis, se considerará como emergencia a todo evento que pueda perjudicar a las personas ocasionado por accidentes de tránsito, accidentes dentro del hogar o algún incidente que genere la necesidad de ser atendido por unidades especializadas.

Luego de realizar un ajuste a la cantidad de emergencias contra los años, se observa que tiene una tendencia creciente en los últimos años, en el Perú, lo cual se puede apreciar en la Figura 10.

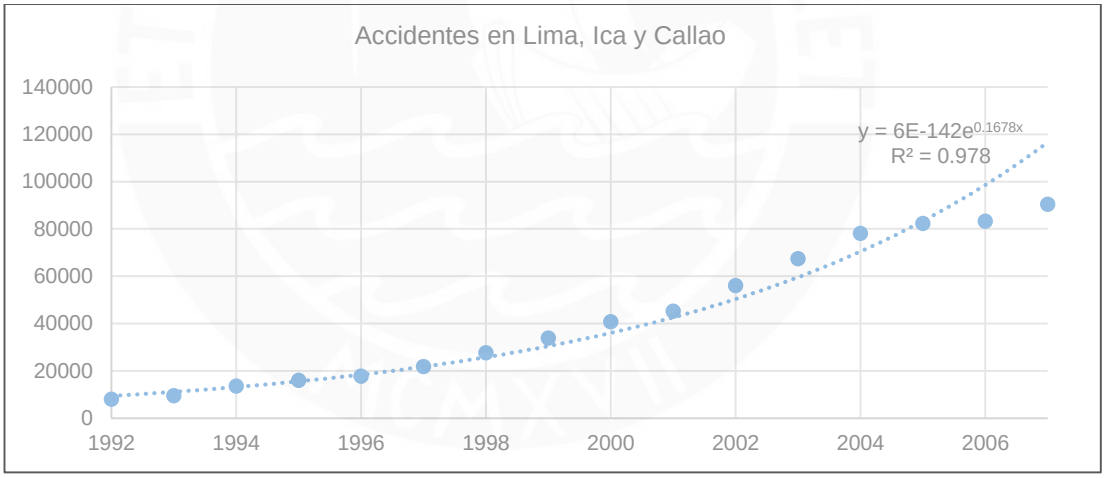



*Figura 10: Tendencia de accidentes en el Perú*
*Elaboración propia*

Existen dos tipos de emergencias, las cuales pueden catalogarse como fatales (aquellas que tienen al menos una persona fallecida) y no fatales. En la Tabla 1 se observa la cantidad de accidentes desde el año 2011 al 2015 en Lima Metropolitana.

*Tabla 1: Accidentes fatales en Lima*

| | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|
| Heridos | 25,212 | 23,133 | 27,640 | 31,996 | 30,573 |
| Muertos | 502 | 496 | 551 | 445 | 417 |

*Fuente: Bases de datos del CGBVP*

De la información de la Tabla 1 se puede inferir que la cantidad de muertos por accidentes fatales ha disminuido en los últimos dos años, pero esta cifra continúa siendo muy alta.

Las emergencias de todo tipo ya sean de tránsito, robos o accidentes en casa, necesitan ser atendidas por ciertas entidades de manera rápida y efectiva para evitar efectos irreversibles. Por ejemplo, en un accidente de tránsito se necesitará la atención inmediata de ambulancias y cuerpo de bomberos dependiendo del grado del accidente. A estas entidades se les considerará dentro del grupo de Unidades de Emergencia. Para fines de la tesis, el análisis se centrará en el Cuerpo General de Bomberos Voluntarios del Perú (CGBVP), y será de vital importancia que estas unidades arriben al lugar del accidente en el menor tiempo posible, ya sea en la vía pública o en propiedad privada, para que los afectados sean atendidos de manera rápida y llevados a los centros de salud más cercanos a la brevedad.

### 3.2. Marco de Referencia

Según MINSA (2008):

- Por cada 24 horas mueren 10 personas por accidentes de tránsito.
- En el 2008 se registraron 85,337 accidentes de tránsito los cuales produjeron 3,489 muertes y 500,059 lesionados a nivel nacional.
- La cantidad de accidentes de tránsito presenta una curva ascendente desde el 2002
- En los últimos diez años han fallecido 32,107 personas y 342,766 han resultado lesionadas a causa de este problema.

Por lo descrito anteriormente, se puede asegurar que los accidentes de tránsito van en aumento y se deben asegurar las medidas necesarias para atenderlas satisfactoriamente.

Además, uno de los principales motivos para atender una emergencia de manera rápida es que, si más se demora en llegar las unidades de emergencia al lugar del hecho, la probabilidad de mortandad del herido aumenta y se distribuye en tres fases:

- Primera fase: Ocurre en los primeros segundos o minutos luego de haber sucedido el accidente. Supone un 10% del total de mortandad.
- Segunda fase: Se producen el 75% de fallecimientos, y se denomina hora de oro, nombre creado por el Dr. Cowley[1], quien lo define como:

"Hay una hora de oro entre la vida y la muerte. Si estás gravemente lesionado, tienes menos de 60 minutos para sobrevivir. Puedes no morir entonces, pero lo puedes hacer tres días o dos semanas después, porque algo ha ocurrido en tu cuerpo que es irreparable."

- Tercera fase: Se producen el 15% de fallecimientos, los cuales ocurren días o semanas después del accidente.

En la Figura 11 se muestra un esquema del índice de mortandad y el tiempo que transcurre al suceder el accidente, lo cual indica que mientras más rápido sea el tiempo de llegada y la atención de las unidades de emergencia, la probabilidad de supervivencia aumenta.

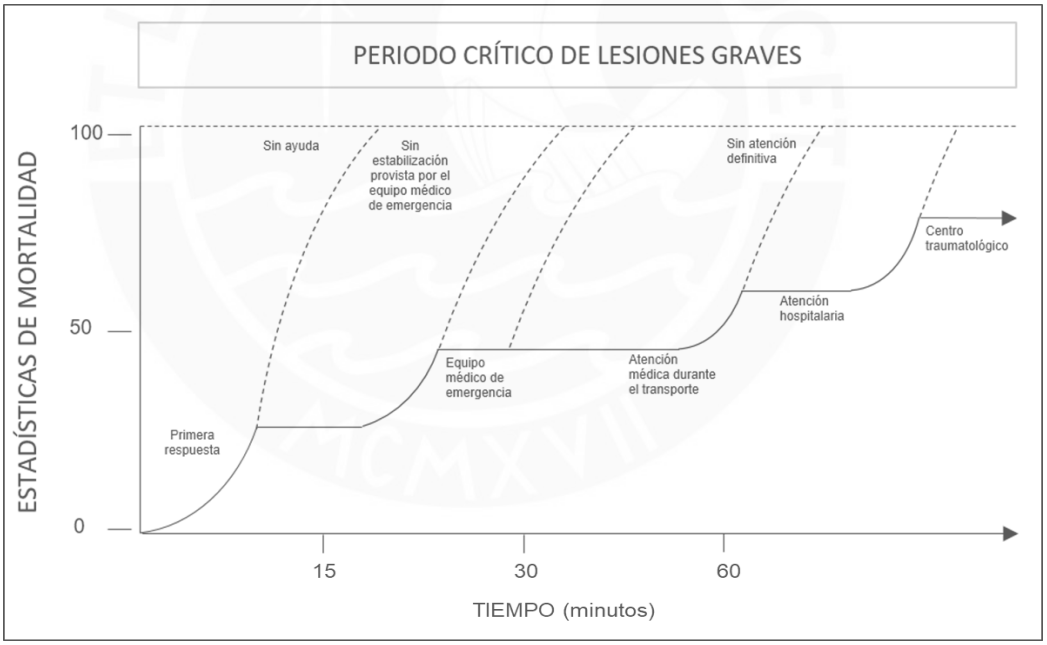



*Figura 11: Periodo crítico de lesiones graves*
*Fuente: The Journal of Medical Society, New Jersey, Vol 74, N°11*

### 3.2.1. Situación actual

**A nivel macro**

Según la Asociación Nacional de Protección contra el Fuego (NFPA), organización encargada de crear y mantener las normas y requisitos mínimos para la prevención

[1] Cirujano militar y director del Centro de Atención al Shock traumático de Maryland (EE.UU.)

contra incendios, el tiempo estándar de llegada de la unidad de emergencia al lugar del accidente debe ser de 4 a 9 minutos como máximo. Este parámetro ha sido logrado por algunos países como Inglaterra, donde el tiempo promedio se llegada a la emergencia es de 8.3 minutos (Tabla 2).

*Tabla 2: Tiempo de respuesta (Inglaterra)*

| TIEMPO DE RESPUESTA EN INGLATERRA (minutos) | | | | | |
|---|---|---|---|---|---|
| **2009-10** | **2010-11** | **2011-12** | **2012-13** | **2013-14** | **2014-15** |
| 8.2 | 8.3 | 8.2 | 8.2 | 8.4 | 8.7 |

*Fuente: Fire & Rescue Statistical Release 19/11/15*

Cabe resaltar que en diversos países están desarrollando algoritmos para poder reducir en tiempo de atención de las ambulancias y diversas unidades de emergencias.

**A nivel Perú**

En el Perú, el tiempo promedio de llegada de la primera unidad de atención de emergencias a un accidente es de 13 minutos, muy por encima del tiempo estandarizado por la NFPA. Esta situación se explica por varias causas, entre las principales, el tráfico denso en la ciudad y la falta de estaciones de bomberos en lugares estratégicos.

Por otro lado, según MINSA (2008) se sabe que la mayoría de los accidentes ocurren entre las 8 a.m. y las 2 p.m., disminuyendo en un 5% con respecto al horario de la tarde a las 8 p.m.; además, los accidentes se producen en mayor proporción los fines de semana de viernes a lunes.

### 3.2.2. Análisis de las causas

Con la ayuda del diagrama causa-efecto de Ishikawa, ubicando como problema principal al elevado tiempo de llegada de una unidad de emergencia al lugar del accidente, se lograron reconocer las principales causas, obteniendo como principales los siguientes:

- Problemas de infraestructura
- Errores humanos
- Errores en tecnología
- Falta y mal uso de recursos
- Desastres naturales

El detalle del análisis en Ishikawa se muestra en la Figura 12.

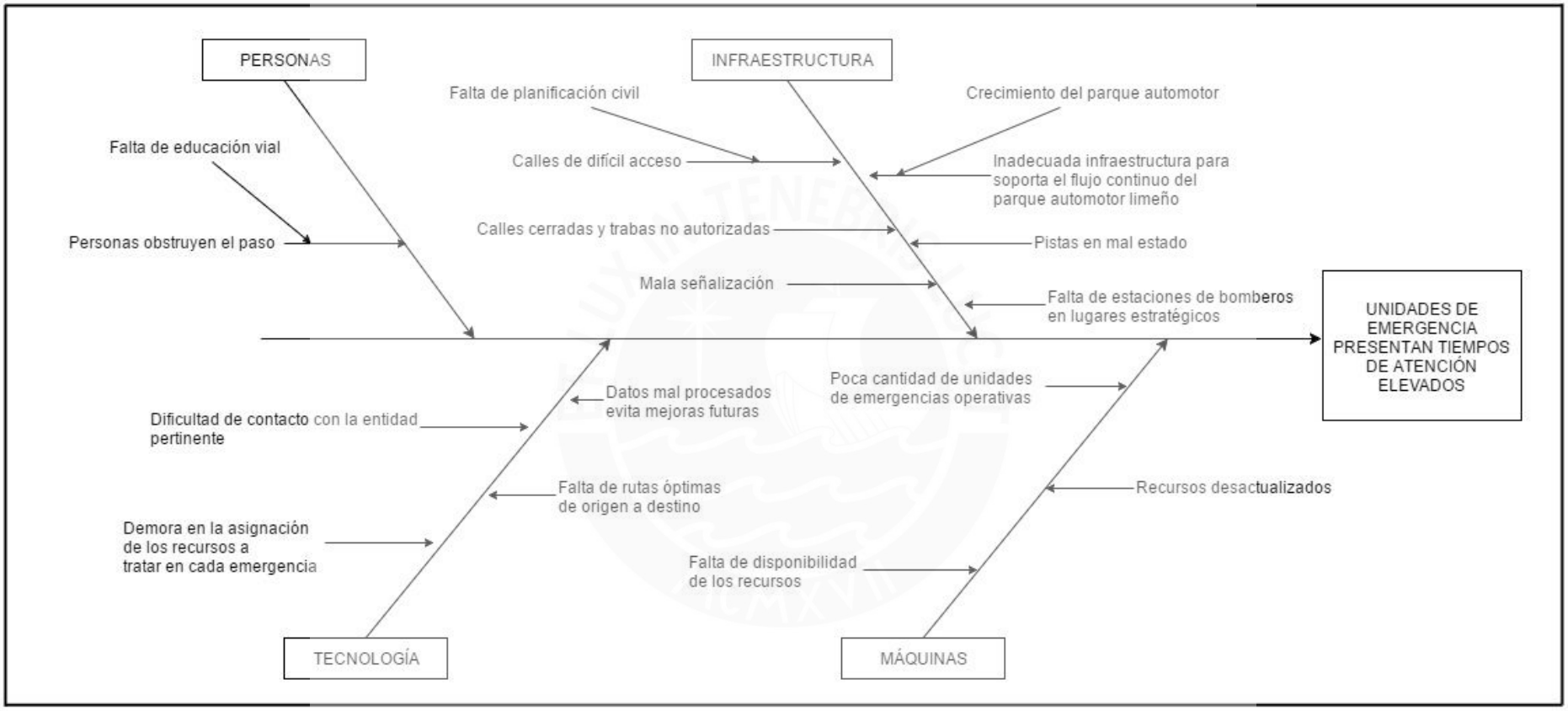



*Figura 12: Diagrama Ishikawa de causas*
*Elaboración propia*

### 3.3. Cuerpo General de Bomberos Voluntarios del Perú

El CGBVP es la institución encargada de prevenir, proteger y brindar apoyo a la población ante posibles desastres, emergencias e incendios. A continuación, se muestra la misión y la visión de la institución extraídas de la página web.

Misión:

> "El Cuerpo General de Bomberos Voluntarios del Perú es la autoridad competente en materia de prevención, control y extinción de incendios, realiza acciones de atención de accidentes vehiculares y emergencias médicas, rescate y salvataje de vidas expuestas a peligro. Brinda sus servicios de manera voluntaria a toda la comunidad debido a su vocación de servicio, sensibilidad social, entrega y disciplina."

Visión:

> "El CGBVP es una Institución consolidada, científica y técnicamente preparada que cumple con su misión, con equipos y maquinarias modernas que permiten un accionar más rápido y efectivo, con personal voluntario capacitado mediante técnicas actualizadas. La difusión de las recomendaciones sobre accidentes y desastres disminuyó el riesgo de siniestros. El ámbito de acción del CGBVP abarca todo el territorio nacional, incluso las zonas que estaban desprotegidas."

Actualmente, el Cuerpo de Bomberos de Lima y Callao se dividen en tres comandancias:

- IV Comandancia departamental Lima: 18 comandancias

*Tabla 3: Lista de comandancias departamentales de Lima*

| IV COMANDANCIA DEPARTAMENTAL LIMA | | | |
|---|---|---|---|
| N° | Nombre | N° Compañía | Dirección |
| 1 | Roma N°2 | 2 | Jr. Junín No. 560 - Cercado de Lima. |
| 2 | France N°3 | 3 | Jr. Moquegua No. 240 - Cercado de Lima |
| 3 | Lima N°4 | 4 | Jr. Manuel Candamo No. 455 |
| 4 | VictoriaN°8 | 8 | Av. Manuel Cisneros No. 597 |
| 5 | Salvadora lima N°10 | 10 | Jr. De La Unión No. 1027 |
| 6 | Internacional N°14 | 14 | Jr. Rebeca Oquendo No. 360 |
| 7 | Rimac N°21 | 21 | Jr. Trujillo No. 836 |
| 8 | Chosica N°32 | 32 | Jr. Callaro No. 168 |
| 9 | Magdalena N°36 | 36 | Av. Sucre No. 899 |
| 10 | San Miguel N°83 | 83 | Jr. Sucre Cdra. 7 - San Miguel |
| 11 | San Isidro N°100 | 100 | Calle Godoffredo García 439, San Isidro |
| 12 | Chaclacayo N°115 | 115 | Av. Nicolas Ayllon Cdra. 15 |
| 13 | San Juan de Lurigancho N° 121 | 121 | Av. El Bosque No. 341, Urb. Canto Grande |
| 14 | Salamanca N°127 | 127 | Las Dalias No. 180 Parque de Monterrico |
| 15 | Técnico Carlos León Delgado N°138 | 138 | Jr. Los Pinos 2da Cdra. Esq Jr. Inclan, Urb. Los Ficus |
| 16 | Cap. CBP Andrés Roman Gutierrez 169 ATE N°169 | 169 | Av. Santa Ana Nro. 940, Puerta 4 Mercado de Productores |
| 17 | El Agustino N°176 | 176 | Jr. Maria Baldarrago s/n Cdra. 5 |
| 18 | Jesús María. Waldo Hernan Olivos V. N°202 | 202 | Jirón Capac Yupanqui, cuadra 16 |

*Elaboración propia*

- XXIV Comandancia departamental Lima Sur: 19 comandancias

*Tabla 4: Lista de comandancias departamentales de Lima Sur*

| XXIV Comandancia Departamental Lima Sur | | | |
|---|---|---|---|
| N° | Compañías | N° Compañía | Dirección |
| 1 | Garibaldi N°6 | 6 | Av. Huaylas No. 298 |
| 2 | Cosmopolita N°11 | 11 | Calle Galeno s/n |
| 3 | Olaya N°13 | 13 | Av. Lima No. 227 |
| 4 | Grau N°16 | 16 | Plaza Espinoza No. 106 |
| 5 | Miraflores N°28 | 28 | Av. Mariscal Caceres No. 172 |
| 6 | Cañete N°49 | 49 | Jr. 28 de Julio No. 809 |
| 7 | La Molina N°96 | 96 | Av. La Chalana/Esq. Av. El Remo S/N S/N Urb. Las Lagunas - La Molina |
| 8 | Villa El Salvador N°105 | 105 | Sector 2, Grupo 15 |
| 9 | Villa María del Triunfo N°106 | 106 | Cdra. 15 Av. Defensores Lima |
| 10 | San Pedro de Mala N°111 | 111 | Escala Baja, Camal Nuevo Antigua Carretera Panamericana |
| 11 | San Juan de Miraflores N°120 | 120 | Esq. Paita con Sullana, Urb. Industrial |
| 12 | Punta Negra N°125 | 125 | Guanay Sur No. 360 |
| 13 | San Pedro de Lurín N°129 | 129 | Jr. Bolívar s/n |
| 14 | Nuestra Señora de La Asunción de María N°133 | 133 | Jr. Huancavelica s/n Chilca Pueblo |
| 15 | Santiago Apóstol N°134 | 134 | Monte de los Olivos Cdra. 9, Urb. Prolog. Benavides 2da Etapa |
| 16 | Nuevo Milenio N°155 | 155 | Av. Atocongo s/n Frente a Cementos Lima |
| 17 | Pachamac N°160 | 160 | Prolog. Los Suspiros Mz. 41 Lt. 12 |
| 18 | Virgen del Carmene N°183 | 183 | AV Benigno Rios N°296 Imperial-Cañete |
| 19 | Brigadier General CBP Roberto Ognio Baluarte N°221 | 221 | AV Benigno Rios N°296 Imperial-Cañete |

*Elaboración propia*

- XXV Comandancia departamental Lima Norte: 13 comandancias

*Tabla 5: Lista de comandancias departamentales de Lima Norte*

| XXV Comandancia Departamental Lima Norte | | | |
|---|---|---|---|
| N° | Compañías | N° Compañía | Dirección |
| 1 | Huacho N°20 | 20 | Jr. Echenique No. 355 |
| 2 | Huaral N°44 | 44 | Calle Los Geranios # 294 esquina con Calle Las Dalias # 151 - Residencial Huaral |
| 3 | San Martín de Porres N°65 | 65 | Jr. San Lucas Nº 401 1era entrada Pallao |
| 4 | Barranca N°73 | 73 | Plaza de Armas s/n |
| 5 | Santiago Távara Renovales N°80 | 80 | 1° De Mayo s/n |
| 6 | Salvadora Paramonga N°81 | 81 | Av. Central 131, Paramonga. |
| 7 | Pativilca - Simón Bolívar N°91 | 91 | Bolívar No. 232 |
| 8 | Comas N°124 | 124 | Manuel Aranguri No. 699 |
| 9 | Puente Piedra N°150 | 150 | José Gálvez No. 315 |
| 10 | Los Olivos N°161 | 161 | Mz. K Lt. 15 3era Etapa El Trébol, Calle 36, Alt. Cdra. 8 Av. Tomas Valle |
| 11 | Ancón N°163 | 163 | Av. Paredez Roncal 755 - Cercado |
| 12 | Carabayllo N°164 | 164 | Jr. No me Olvides s/n, Urb. Santa Isabel |
| 13 | Independencia N°168 | 168 | Jr. Los Tumbos 1ra Cdra. S/N Espalda de la Municipalidad |
| 14 | San Jerónimo de Sayán N°216 | 216 | Compañía Nueva |

*Elaboración propia*

*Tabla 6: Lista de comandancias departamentales Callao*

| V COMANDANCIA DEPARTAMENTAL CALLAO | | | |
|---|---|---|---|
| N° | Nombre | N° Compañía | Dirección |
| 1 | Unión Chalaca N°1 | 1 | Av. Dos de Mayo No. 375 - Cercado del Callao |
| 2 | Italia Nº5 | 5 | Av. Alejandro Granda s/n Mz. E Lt. 3, Urb. Stella Maris |
| 3 | Garibaldi Nº7 | 7 | Av. La Marina Cdra. 40 |
| 4 | Salvadora Callao Nº9 | 9 | Jr. Pedro Ruiz Gallo No. 246 |
| 5 | Callao Nº15 | 15 | Jr. Los Heros No. 151 |
| 6 | Perú Nº18 | 18 | Jr. Puno No. 137 |
| 7 | La Punta Nº34 | 34 | Av. Bolognesi No. 133 |
| 8 | Antonio Alarco Espinoza Nº60 | 60 | Jr. Cóndores No. 591 |
| 9 | Técnico Lorenzo Giraldo Vega Nº75 | 75 | Pedro Beltrán s/n |
| 10 | BRIGADIER CBP ALEJANDRO REYES LEON Nº184 | 184 | Nuestra Señora de las Mercedes - Mi Perú |
| 11 | Adolfo Martín Link Leoane N°207 | 207 | Av. Panamá Mz. "K", Lote 1 "A" del AA HH Nuevo Progreso - Ventanilla |

*Elaboración propia*

En la Figura 13 se muestra la distribución de estaciones pertenecientes a las comandancias IV, V, XXIV y XXV dentro del mapa de Lima.

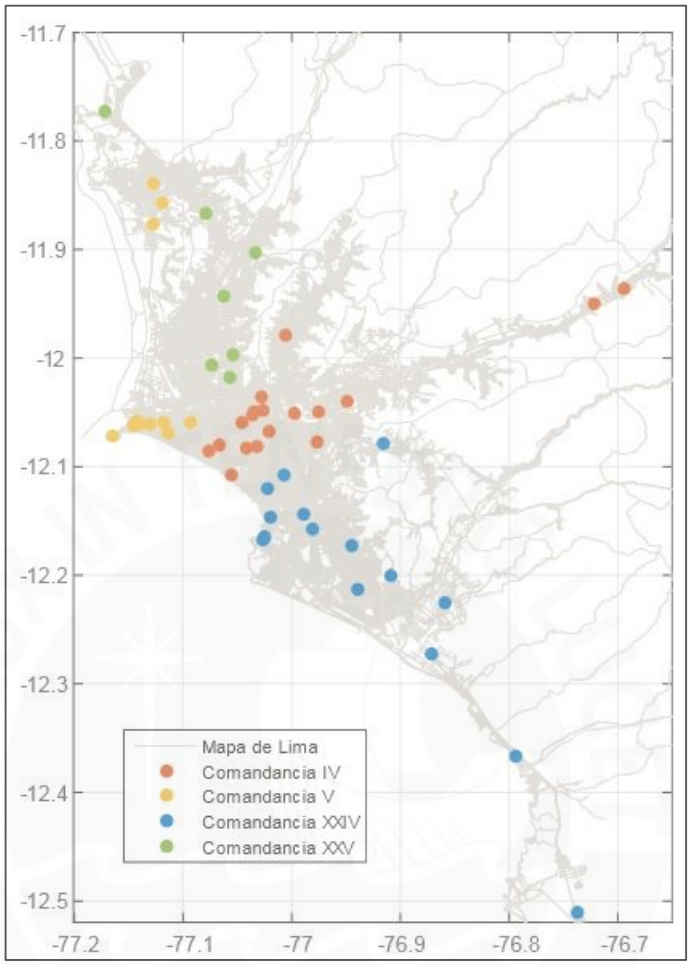



*Figura 13: Distribución de comandancias en Lima*
*Elaboración propia*

### 3.3.1. Los recursos

Los recursos con los que cuenta el CGBVP se clasificarán en personal, vehículos y sistemas.

#### 3.3.1.1. Personal

El personal perteneciente al CGBVP se clasificará en dos tipos:

Personal de bomberos

- Bombero al mando de la unidad
- Piloto del vehículo
- Serumistas
- Comandante de la emergencia

Personal de oficina

- Operadores de la central de emergencia (CEEM)
- Supervisores de la CEEM
- Personal de dirección de informática
- Director de comunicaciones

### 3.3.1.2. Vehículos

Cada vehículo perteneciente a una estación de bomberos cuenta con una codificación asignada por la Dirección General de Operaciones (DIGO), donde se le asigna una abreviación según el tipo de vehículo, el número de compañía y otras características especiales con las que cuente. Si una estación cuenta con dos vehículos o más del mismo tipo, se le colocará un número correlativo al final del código comenzando por uno.

Los vehículos se dividirán en dos categorías: vehículos contra incendios y vehículos especiales/otros.

- Vehículos contra incendios

*Tabla 7: Lista de vehículos contra incendios*

| VEHÍCULO | ABREVIACIÓN | CÓDIGO ADICIONAL | CODIFICACIÓN |
|---|---|---|---|
| Autobomba | M | Cod 1: Capacidad de 1000 a 1500 galones. | MB"N° de base donde pertenece el vehículo"-"Código adicional". Ejemplo: M15-1, es una autobomba (máquina) que pertenece a B15 y tiene una capacidad de 1000 galones de agua. |
| | | Cod 2: Capacidad de 750 a 999 galones. | |
| | | Cod 3: Capacidad de 500 a 749 galones. | |
| | | Cod 4: Capacidad de 50 a 449 galones. | |
| | | Cod 5: Generado de espuma con bomba de concentrado. | |
| | | Cod 6: Autobombas sin tanque. | |
| Ambulancia | AMB | - | AMB - "N° de base donde pertenece el vehículo". Ejemplo: AMB-10, es una ambulancia que pertenece a B10. |
| Médica | MED | - | MED - "N° de base donde pertenece el vehículo". Ejemplo: MED-28, es una médica que pertenece a B28. |
| Rescate pesado | RES | - | RES - "N° de base donde pertenece el vehículo". Ejemplo: RES-6, es la unidad de rescate que pertenece a B6. |
| Rescate ligero | | | |
| Brazo articulado | SNOR | - | SNOR - "N° de base donde pertenece el vehículo". Ejemplo: SNOR-4, es la unidad aérea con brazo articulado de B4. |
| Escala | ESC | - | ESC - "N° de base donde pertenece el vehículo". |

| | | | |
|---|---|---|---|
| | | | Ejemplo: ESC-10, es la unidad aérea con escala que pertenece a B10. |

*Elaboración propia*

- Vehículos especiales / otros:

*Tabla 8: Lista de Vehículos Especiales y Otros*

| VEHÍCULO | ABREVIACIÓN | CÓDIGO ADICIONAL | CODIFICACIÓN |
|---|---|---|---|
| Materiales peligrosos | MAPEL | - | MAPEL - "N° de base donde pertenece el vehículo". Ejemplo: MAPEL-28, es la unidad de materiales peligrosos que pertenece a B28. |
| Eléctrica | ELEC | - | ELEC - "N° de base donde pertenece el vehículo". Ejemplo: ELEC-36, es la unidad eléctrica que pertenece a B36. |
| Cisterna | CIST | - | CIST - "N° de base donde pertenece el vehículo". Ejemplo: CIST-18, es una unidad cisterna de B18 con 3,000 galones de agua. |
| Médica masiva | UAM | - | UAM-XXX, es la unidad de apoyo masivo. |
| UNIMOG | UNIMOG | - | UNIMOG-XXX, vehículo con tracción en las 4 ruedas. |
| USAC | USAC | - | USAC-XXX, unidad de soporte para aire autocontenido. |
| BREC | BREC | - | BREC, unidad de búsqueda para rescate en estructuras colapsadas. |
| Auxiliares | AUX | - | AUX - "N° de base donde pertenece el vehículo". Ejemplo: AUX-4, es la unidad auxiliar de la compañía que pertenece a B36. |
| Ómnibus | BUS | - | BUS - "N° de base donde pertenece el vehículo". Ejemplo: BUS-18, ómnibus que pertenece a B18. |

*Elaboración propia*

Es importante aclarar que los vehículos pertenecientes a cada estación de bomberos no rotan de compañía, por lo tanto, luego de cada salida regresa a su lugar de origen.

Para el caso de estudio, solo se utilizarán 13 tipos de vehículos del total de lista y se clasifican según la Tabla 9.

*Tabla 9: Lista de vehículos considerados en el modelo*

| N° | COD | Tipo de Vehículo |
|---|---|---|
| **1** | AMB | Ambulancia |
| **2** | MED | Médica |
| **3** | RES | Rescate |
| **4** | MAT | Materiales Peligrosos |
| **5** | M | Autobomba |
| **6** | CIS | Cisterna |
| **7** | AUX | Auxiliares |
| **8** | SNO | Brazo articulado |
| **9** | USA | Unidad de soporte para aire autocontenido |
| **10** | ESC | Escala |
| **11** | ELE | Eléctrica |
| **12** | SAM | SAMU |
| **13** | QUI | Química |

*Elaboración propia*

#### 3.3.1.3. Sistemas

Los sistemas del CGBVP se clasificarán en tres.

- Asistencia del personal

Cada ingreso y salida del personal de bomberos es registrado desde un terminal de cómputo a través de la intranet para que la central de emergencia pueda tener información a tiempo real de la disponibilidad de cada estación. En esta etapa, el bombero voluntario registra el servicio que brindará: piloto voluntario, médico, paramédico, técnico en materiales peligrosos y/o especialista en búsqueda y rescate en espacios confinados.

- Sistema computarizado CEEM

Este es el sistema que utilizan los operadores para registrar los partes de las emergencias, los datos necesarios, revisan la disponibilidad de los recursos como vehículos y personal, ingresan el tipo de emergencia, nombre del informante, dirección del lugar de destino y generan las órdenes para la atención.

- Sistema de SISEMER

El manual de usuario del sistema central de emergencias (SISEMER) es donde se guarda toda la información obtenida del sistema CEEM. Por otro lado, cuenta con la información de los protocolos, la base de datos de todos los partes, de los tipos de

emergencias y es utilizado para poder ubicar geográficamente a la emergencia y lo muestra en el mapa cartográfico de Lima.

### 3.3.2. Tipos de emergencias

El Cuerpo General de Bomberos Voluntarios del Perú ha clasificado las emergencias en siete grupos:

- Incendio

Comprende a los accidentes cuya consecuencia conlleve al incendio de estructuras, vehículos, establecimientos, equipos, etc.

- Materiales peligrosos

Comprende a los accidentes que sean ocasionados por figa de materiales inflamables como gases, hidrocarburos, productos químicos, pirotécnicos, etc.

- Emergencia médica

Comprende a los accidentes que impliquen traslado de emergencia, malestares de salud generales en las personas y traumas físicos.

- Rescate

Comprende a los rescates en acantilados, derrumbes, ascensores, etc.

- Accidentes vehiculares

Comprende a los accidentes en vehículos particulares, transporte público, urbano, privado, de carga, etc.

- Desastres naturales

Comprende a los accidentes ocasionado en inundaciones, huaycos, sismos, etc.

- Servicios especiales

Comprende a las charlas de capacitación, eventos públicos, eventos oficiales, etc.

Las emergencias, subtipos y códigos se detallan de la Tabla 10 a la Tabla 16, los cuales fueron extraídos de la información brindada por el CGBVP.

### 3.3.2.1. Incendio

*Tabla 10: Subtipos de emergencia tipo 1*

| | | | | |
|---|---|---|---|---|
| **INCENDIO (01)** | **Estructuras (0101)** | Vivienda (010101) | Material noble | 01010101 |
| | | | Material precario | 01010102 |
| | | Edificio (010102) | Entre 1 al 6 | 01010201 |
| | | | Entre 7 al 11 | 01010203 |
| | | | Entre 12 a más | 01010204 |
| | | Planta de producción | | 010103 |
| | | Almacén - depósito | | 010104 |
| | | Centro comercial, autoservicios y galerías | | 010105 |
| | | Sótanos | | 010106 |
| | | Centros de estudios | | 010107 |
| | | Centros de reclusión | | 010108 |
| | **Vehículo (0102)** | Automóvil (010201) | Cochera - garaje - taller | 01020101 |
| | | | Vía pública | 01020102 |
| | | Combi - coaster (010203) | Cochera - garaje - taller | 0102001 |
| | | | Vía pública | 01020302 |
| | | Camión con carga (010204) | Cochera - garaje - taller | 01020401 |
| | | | Vía pública | 01020402 |
| | | Camión sin carga (010205) | Cochera - garaje - taller | 01020501 |
| | | | Vía pública | 01020502 |
| | | Ómnibus (010206) | Cochera - garaje - taller | 01020601 |
| | | | Vía pública | 01020602 |
| | **Unidades de atención hospitalaria y de reposo (0103)** | Hospital y clínica | | 010301 |
| | | Asilos | | 010301 |
| | | Centro médico y posta | | 010303 |
| | **Equipos eléctricos con energía (0104)** | Sub-estación de alta tensión | | 010401 |
| | | Postes | | 010402 |
| | | Letreros luminosos | | 010403 |
| | | Transformadores de alta tensión | | 010404 |
| | | Sub-estación de transmisión | | 010405 |
| | | Caída de cable de alta tensión | | 010406 |
| | **Forestal (0105)** | Bosques | | 010501 |
| | | Parque zonal | | 010502 |
| | | Terreno de cultivo - chacra - otros | | 010503 |
| | **Terreno baldío urbano (0106)** | | | 0106 |
| | **Otros (0107)** | Embarcaciones en tierra (astilleros) | | 010701 |
| | | Establos y granjas | | 010702 |
| | | Puerto marítimo | | 010703 |

*Elaboración propia*

### 3.3.2.2. Materiales peligrosos

*Tabla 11: Subtipos de emergencias tipo 2*

| | | | | |
|---|---|---|---|---|
| **MATERIALES PELIGROSOS (02)** | **Fuga de GLP y otros gases inflamables (0201)** | Balón domiciliario (10 Kg) | | 020101 |
| | | Balón 100 lb (46 Kg) (020102) | Vehículo | 02010207 |
| | | Tanque estacionario (250 a 1000 gal) (020103) | Vivienda | 02010301 |
| | | | Edificio entre 1 al 6 | 02010302 |
| | | | Edificio entre 7 al 11 | 02010304 |
| | | | Edificio entre 12 a más | 02010305 |
| | | | Centro comercial | 02010306 |
| | | | Industria (más de 1000 gal) | 02010307 |
| | | Camión cisterna de gas (020104) | Vía pública | 02010401 |
| | | | Cochera - garaje - taller | 02010402 |
| | | | Industria | 02010403 |
| | | Estación de servicio | | 020105 |
| | | Centro de distribución | | 020106 |

| | | | | |
|---|---|---|---|---|
| | | Planta de envasado | | 020107 |
| | | Planta de distribución | | 020108 |
| | **Hidrocarburos (0202)** | Camión cisterna de gas (020201) | Vía pública | 02020101 |
| | | | Cochera - garaje - taller | 02020102 |
| | | | Estación de servicio | 02020103 |
| | | | Industria | 02020104 |
| | | Estación de servicio | | 020202 |
| | | Planta de almacenamiento | | 020203 |
| | | Refinería | | 020204 |
| | **Productos químicos (0203)** | Camión cisterna (020301) | Vía pública | 02030101 |
| | | | Cochera - garaje - taller | 02030102 |
| | | | Industria | 02030103 |
| | | Contenderoes (cilin., sacos, bot.) | | 020302 |
| | | Industrias | | 020303 |
| | **Material pirotécnico - explosivos (incendio) (0204)** | Vivienda | | 020401 |
| | | Fábrica | | 020402 |
| | | Almacén | | 020403 |
| | | Centro comercial | | 020404 |
| | | Clandestino | | 020405 |
| | | Vía pública | | 020406 |
| | **Otros (0205)** | Material radiactivo 1 | | 020501 |
| | | Material radiactivo 2 | | 020502 |
| | **Presencia posible de artefacto explosivo** | | | 0206 |

*Elaboración propia*

### 3.3.2.3. Emergencia Médica

*Tabla 12: Subtipos de emergencias Tipo 3*

| | | | |
|---|---|---|---|
| **EMERGENCIA MÉDICA (03)** | **Tipo médico (0301)** | Dolor abdominal | 030101 |
| | | Reacción alérgica | 030102 |
| | | Dificultad respiratoria | 030103 |
| | | Dolor de pecho | 030104 |
| | | Convulsiones | 030105 |
| | | Problema en diabéticos | 030106 |
| | | Dolor de cabeza | 030107 |
| | | Problemas cardiacos | 030108 |
| | | Envenenamiento - sobredosis | 030109 |
| | | Problemas psiquiátricos | 030110 |
| | | Paciente enfermo | 030111 |
| | | Derrame cerebral - ACV | 030112 |
| | **Traumáticas (0302)** | Mordedura de animal | 030201 |
| | | Asalto - agresión | 030202 |
| | | Quemadura | 030203 |
| | | Lesión - problema ocular | 030204 |
| | | Herido por caída | 030205 |
| | | Exposición a frío o calor | 030206 |
| | | Hemorragia | 030207 |
| | | Accidente industrial | 030208 |
| | | Herido por arma de fuego | 030209 |
| | | Herido por arma blanca | 030210 |
| | | Herido por atropello | 030211 |
| | **Eventos críticos tiempo - vida (0303)** | Inhalación gases tóxicos | 030301 |
| | | Paro cardiaco | 030302 |
| | | Atragantamiento | 030303 |
| | | Ahogamiento | 030304 |
| | | Electrocutado | 030305 |
| | | Embarazo - parto | 030306 |
| | | Inconsciente - desmayo | 030307 |
| | **Traslados** | | 0304 |

*Elaboración propia*

### 3.3.2.4. Rescates

*Tabla 13: Subtipos de emergencias tipo 4*

| | | | | |
|---|---|---|---|---|
| **RESCATES (04)** | **Acantilado (0401)** | Con acceso inferior | | 040101 |
| | | Sin acceso inferior | | 040102 |
| | **Derrumbe (0402)** | Con personas atrapadas (040201) | Primer piso | 04020101 |
| | | | Segundo piso a más | 04020102 |
| | | Sin personas atrapadas | | 040202 |
| | **Otros (0403)** | En mar | | 040301 |
| | | En río | | 040302 |
| | | En tanque de agua aéreo | | 040303 |
| | | En tanque de agua subterráneo | | 040304 |
| | | En silo letrinas | | 040305 |
| | | Alcantarillado | | 040306 |
| | | Lago lagunas | | 040307 |
| | | Torre de alta tensión | | 040308 |
| | **Ascensor (0404)** | Personas atrapadas | | 040401 |
| | **Aeronaves (0405)** | Comercial | | 040501 |
| | | Militar | | 040502 |
| | | Helicóptero | | 040503 |
| | **Trenes (0406)** | Con carga | | 040601 |
| | | Con pasajeros | | 040602 |
| | **Suicida (0407)** | En altura | | 040701 |
| | | Encerrado en habitación | | 040702 |

*Elaboración propia*

### 3.3.2.5. Accidente vehicular

*Tabla 14: Subtipos de emergencias tipo 5*

| | | | |
|---|---|---|---|
| **ACCIDENTE VEHICULAR (05)** | **Particular (0501)** | Automóvil | 050101 |
| | | Camioneta | 050102 |
| | | Moto / Mototaxi | 050103 |
| | **Transporte escolar (0502)** | Combi | 050201 |
| | | Coaster | 050202 |
| | | Ómnibus | 050203 |
| | **Transporte pasajeros urbano (0503)** | Combi | 050301 |
| | | Coaster | 050302 |
| | | Ómnibus | 050303 |
| | **Transporte pasajeros interprovincial (0504)** | Camión con pasajeros | 050401 |
| | | Ómnibus | 050402 |
| | **Transporte de carga (0505)** | Cisterna | 050501 |
| | | Camión | 050502 |
| | | Trailer | 050503 |
| | **Vehículo CBP** | | 0506 |
| | **Vehículo oficial** | | 0507 |

*Elaboración propia*

### 3.3.2.6. Desastres naturales

*Tabla 15: Subtipos de emergencias tipo 6*

| | | |
|---|---|---|
| **DESASTRES NATURALES (06)** | **Inundación** | 0601 |
| | **Huayco** | 0602 |
| | **Sismos** | 0603 |
| | **Maretazos** | 0604 |
| | **Otros** | 0699 |

*Elaboración propia*

### 3.3.2.7. Servicio especial

*Tabla 16: Subtipos de emergencias tipo 7*

| | | | |
|---|---|---|---|
| **SERVICIO ESPECIAL (07)** | **Prevención - capacitación (0701)** | Charlas - cursos CGBVP | 070101 |
| | | Prácticas | 070102 |
| | | Simulacros | 070103 |
| | | Charlas - cursos (particular) | 070104 |
| | **Eventos públicos (0702)** | Campañas de vacunación | 070201 |
| | | Agua a instituciones | 070202 |
| | | Conciertos | 070203 |
| | | Festivales | 070204 |
| | | Ferias | 070205 |
| | | Otros | 070206 |
| | **Eventos oficiales (0703)** | Ceremonias | 070301 |
| | | Velatorios - entierros | 070302 |
| | | Otros | 070303 |
| | **Otros eventos (0704)** | Abrir puertas | 070401 |
| | | Colocar driza, letreros, otros | 070402 |

*Elaboración propia*

Por otro lado, el tipo de emergencia con mayor número de ocurrencias es la médica, seguido por incendios y accidentes vehiculares.

*Tabla 17: Cantidad relativa por tipo de emergencia*

| Tipo | Tipo de emergencia | Cantidad | % |
|---|---|---|---|
| 1 | Incendio | 7,867 | 9% |
| 2 | Materiales peligrosos (incidente) | 4,324 | 5% |
| 3 | Emergencia médica | 50,222 | 72% |
| 4 | Rescate | 1,664 | 2% |
| 5 | Accidente vehicular | 8,551 | 9% |
| 6 | Desastres naturales | 31 | 0% |
| 7 | Servicio especial | 2,009 | 3% |
| | **Total general** | **72,671** | **100%** |

*Elaboración propia*

### 3.3.3. Protocolos actuales

A continuación, se presenta un breve resumen de los protocolos del CGBVP, cuya información fue brindada por la misma institución.

- Procedimiento ejecutivo de la recepción de la llamada de emergencia

Este proceso comienza cuando el operador contesta la llamada realizada por un personal perteneciente al cuerpo de bomberos o una persona ajena a la compañía, quien comunica el accidente ocurrido, la hora, la dirección de referencia, el nombre del informante, el tipo de emergencia (principal o secundaria) y el teléfono del informante. Cuando el operador obtiene la dirección de la emergencia, ingresa los datos en el sistema CEEM, el cual está conectado al SISEMER, donde se obtiene la ubicación geográfica de la emergencia y muestra la posición en el mapa. Luego del ingreso de datos, se le indica al informante que se está procediendo al despacho de las unidades y se genera un número correlativo de parte.

- Procedimiento ejecutivo del despacho de unidades

Con la información recopilada en el primer procedimiento (dirección y tipo de emergencia), el sistema CEEM muestra las unidades a despachar de acuerdo con los protocolos de despacho establecidos y el operador envía obligatoriamente las unidades. Como paso siguiente, el operador activa la comunicación mediante radios con la estación donde se encuentran los vehículos a despachar y brinda la información del tipo de accidente, la denominación del vehículo y la dirección de destino. Cuando la estación recibe esta información, el bombero encargado debe responder en un lapso menor a 30 segundos con la confirmación, en caso contrario, se asignará la emergencia a otra estación. Al realizar la salida del vehículo desde la estación hacia el lugar del accidente, se debe informar al operador la hora de salida, la identificación y el número de efectivos presentes en la unidad. Si el operador dentro del sistema observa que un vehículo aparece como disponible, pero se le informa que no lo está, debe cambiar su estado a fuera de servicio. Toda esta información se registra en el SISEMER. Cuando la unidad despachada llega al lugar del accidente, debe comunicar la hora exacta al operador que lo contactó.

- Procedimiento ejecutivo de personal en caso de emergencia

El personal enviado no debe llamar a los operadores de la compañía para consultas, solicitud de información, reclamos, etc. Solo los jefes de brigada pueden comunicarse en caso de emergencias mayores.

- Procedimiento ejecutivo del progreso de la emergencia

El primer efectivo en llegar al accidente debe reportar el primer escenario al CEEM, también debe detallar el área involucrada, el riesgo para vidas y el riesgo de propagación. En el caso de accidentes vehiculares, debe detallar la información de los tipos de vehículos involucrados, número de placas, heridos y fallecidos. Luego, se debe clasificar la emergencia y notificar al CEEM para que se despachen vehículos de acuerdo con el protocolo cuando hagan falta. En el caso que se necesite enviar heridos a los nosocomios, se tendrá que informar el número de pacientes, nombre del nosocomio, la hora de llegada y hora de retirada de la unidad. Estos datos se guardan en el sistema SISEMER. Por último, se debe reportar al CEEM la hora donde la emergencia esté controlada.

- Procedimiento ejecutivo del cierre de emergencia

En este procedimiento, el comandante de la emergencia da la orden de retirada parcial o total de las unidades de emergencia, informando al CEEM e ingresando la

información al SISEMER la hora de salida del accidente y la hora de llegada al cuartel. Luego, el bombero al mando de la emergencia ingresa los datos reales del accidente, como son el tipo de emergencia real, el nombre del bombero al mando, nombres de efectivos que asistieron al accidente, nombre del piloto, tipo del inmueble, daños observados, trabajo efectuado, material utilizado, observaciones, vehículos de otras instituciones involucrados (PNP, Serenazgo, etc), nombre de los pacientes, edades, DNI y observaciones adicionales. Estos datos deben ser llenados dentro de los 30 minutos luego de ingresar a la estación, y luego se procede a cerrar el parte de la emergencia.

- Procedimiento ejecutivo de estado de compañías

En este procedimiento, se registran los estados de los vehículos de la compañía. El bombero al mando de la compañía debe modificar los estados de sus unidades desde la intranet cuando se produzcan la llegada o retiro de un piloto, llegada o retiro del personal y llegada o salida del personal de salud. Además, se debe registrar el estado de los vehículos que esta fuera de servicio por desperfectos mecánicos o por falta de personal, piloto o combustible.

- Procedimiento ejecutivo de unidades de comisión

Este procedimiento comprende a todas las salidas del personal de bomberos o de vehículos por motivos diferentes a emergencias, como encargos de envío de documentos, reparación o mantenimiento de vehículos, etc. Todo cambio de estado debe ser documentado en el SISEMER para que esta información se refleje a tiempo real y no sean incluidos en una emergencia.

### 3.3.4. Principales indicadores

A continuación, se mostrarán los principales indicadores necesarios para poder desarrollar el modelo de la tesis y obtener un mejor análisis de la situación:

- Tiempo promedio de atención a emergencias
- Desviación estándar del tiempo de atención a emergencias
- Número de vehículos per cápita
- Número de emergencias per cápita (índice de frecuencia)
- Número de accidentes por año (índice de frecuencia)
- Indicadores operativos de eficacia
    - Utilización de vehículos por estación
    - Tiempo para la asignación de recursos
    - Tiempo de movilización del recurso (demora en salir de su origen)

- Tiempo de respuesta (llegar al lugar)

### 3.3.5. Flujograma del proceso

A continuación, se detalla el flujograma del proceso de atención a emergencia por parte del CGBVP.

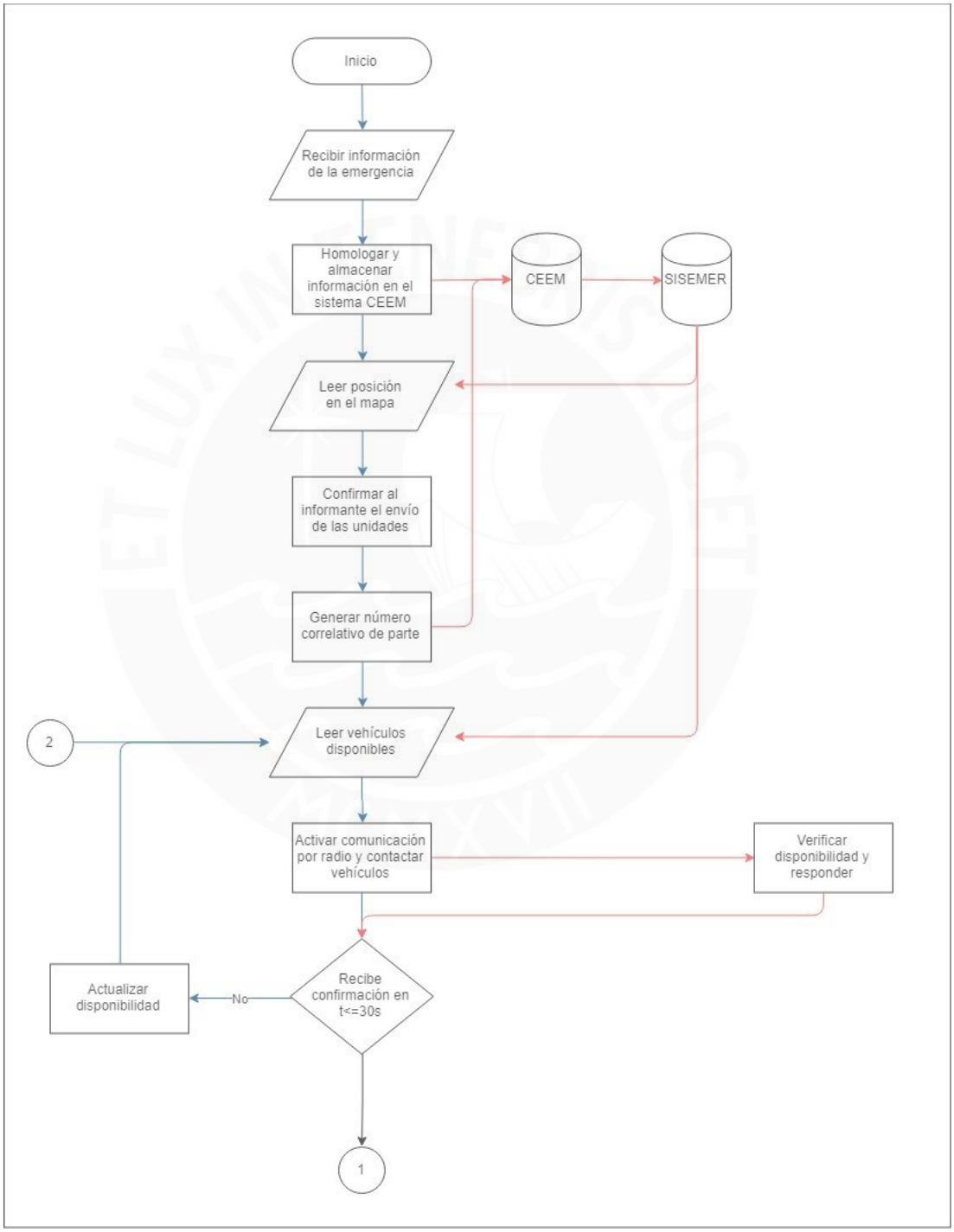



*Figura 14: Flujograma del proceso del CGBVP (parte 1)*

*Fuente: CGBVP*

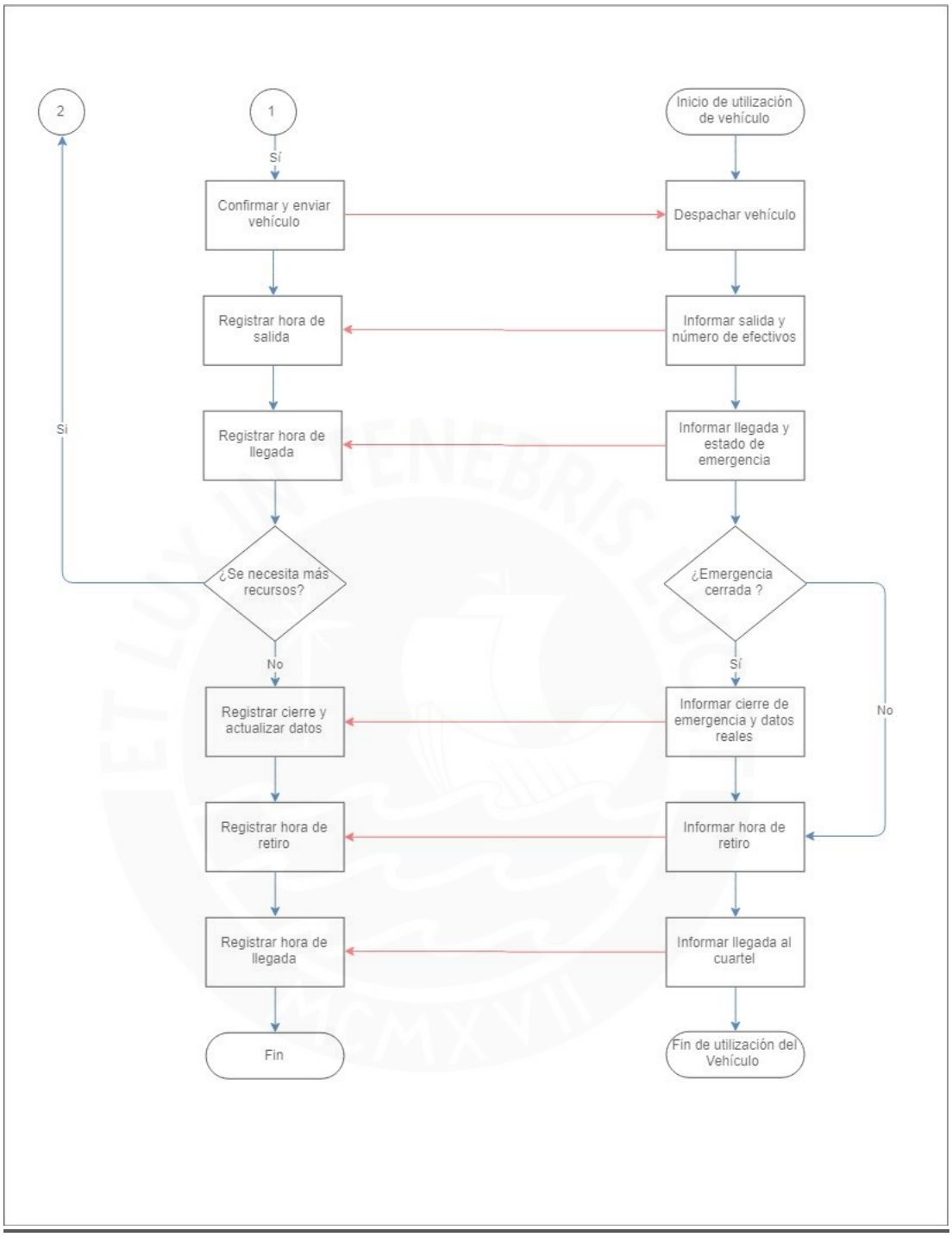



*Figura 15: Flujograma del proceso del CGBVP (parte 2)*

*Fuente: CGBVP*

### 3.3.6. Frecuencia y distribución de emergencias

Luego de realizar el análisis de la data brindada por el CGBVP se obtiene que los distritos con mayor número de emergencias son Cercado de Lima, Santiago de Surco y el Callao.

*Tabla 18: Cantidad de emergencias por tipo y año*

| DISTRITO | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|
| LIMA | 4,309 | 4,146 | 4,713 | 5,523 | 5,275 |
| SANTIAGO DE SURCO | 3,221 | 3,319 | 3,649 | 3,979 | 3,485 |
| CALLAO | 3,806 | 2,708 | 2,783 | 3,467 | 3,645 |
| SAN MARTIN DE PORRES | 1,946 | 2,457 | 2,949 | 3,089 | 2,985 |
| CHORRILLOS | 1,878 | 1,717 | 2,398 | 2,646 | 2,297 |
| LA VICTORIA | 1,955 | 2,009 | 2,123 | 2,462 | 2,370 |
| MIRAFLORES | 1,906 | 1,957 | 2,185 | 2,304 | 2,291 |
| LA MOLINA | 1,746 | 1,860 | 1,996 | 2,122 | 1,721 |
| COMAS | 1,146 | 1,473 | 2,042 | 2,421 | 2,008 |
| SAN JUAN DE LURIGANCHO | 1,345 | 1,773 | 2,004 | 2,065 | 1,893 |
| SAN JUAN DE MIRAFLORES | 1,656 | 1,465 | 1,946 | 1,904 | 1,865 |
| SAN ISIDRO | 1,574 | 1,578 | 1,647 | 1,976 | 1,861 |
| SAN MIGUEL | 1,600 | 1,481 | 1,534 | 1,823 | 1,758 |
| LOS OLIVOS | 1,197 | 1,477 | 1,775 | 1,931 | 1,766 |
| RIMAC | 1,321 | 1,517 | 1,573 | 1,849 | 1,817 |
| SAN BORJA | 1,645 | 1,435 | 1,520 | 1,737 | 1,673 |
| ATE | 1,567 | 1,317 | 1,416 | 1,649 | 1,587 |
| VILLA MARIA DEL TRIUNFO | 1,415 | 1,200 | 1,582 | 1,670 | 1,526 |
| JESUS MARIA | 1,204 | 1,075 | 1,219 | 1,670 | 1,836 |
| VILLA EL SALVADOR | 1,177 | 1,223 | 1,440 | 1,581 | 1,557 |
| SURQUILLO | 1,281 | 1,174 | 1,172 | 1,479 | 1,472 |
| ICA | 901 | 905 | 1,240 | 1,498 | 1,878 |
| BREÑA | 1,156 | 1,164 | 1,235 | 1,384 | 1,482 |
| PUEBLO LIBRE | 1,021 | 1,076 | 1,196 | 1,384 | 1,364 |
| LINCE | 1,016 | 983 | 1,040 | 1,373 | 1,320 |
| HUACHO | 807 | 903 | 1,287 | 1,464 | 786 |
| BARRANCO | 1,036 | 888 | 1,006 | 1,247 | 1,020 |
| MAGDALENA DEL MAR | 811 | 893 | 881 | 1,159 | 1,181 |
| INDEPENDENCIA | 745 | 1,046 | 928 | 1,114 | 954 |
| OTROS | 11,669 | 11,186 | 12,170 | 14,309 | 15,998 |

*Elaboración propia*

# CAPÍTULO 4. APLICACIÓN DE LA METODOLOGÍA PROPUESTA

El primer paso por seguir en la elaboración del modelo es la depuración de los datos, debido a que la información recopilada en la mayoría de los casos no se encuentra estructurada para ser explotada directamente. Posteriormente, se plantea el tratamiento de estos datos donde se transforman y se estructuran para un procesamiento eficiente. Además, para seguir los lineamientos de eficiencia, se pre procesan algunos datos estáticos como las matrices de distancias y de rutas.

Uno de los factores importantes dentro del modelo es el tráfico, el cual es definido como las unidades de tiempo necesarios para avanzar una unidad de distancia. En el modelo propuesto se procede a estimar el tráfico mediante varias distribuciones correspondientes a cada bloque horario. Cabe resaltar que, debido a la ausencia directa de esta información, el tráfico se calcula con la data histórica de los tiempos de demora en llegadas y retornos de los vehículos a las emergencias.

Con toda la información ya procesada, se realizará como siguiente paso la simulación del sistema. Esta etapa se divide en dos subetapas: la presimulación y la simulación de emergencias, las cuales se realizarán de esta manera ya que existen variables estocásticas independientes al estado del modelo y su simulación se puede realizar en lotes; para las demás variables, su valor se determina durante el proceso de la simulación propiamente dicha.

Por último, el modelo es validado mediante distintas pruebas estadísticas que permiten discernir la veracidad de los resultados.

## 4.1. Funciones básicas

Durante el desarrollo de los algoritmos de la tesis se hará uso de funciones que serán expuestas a continuación.

### 4.1.1. Borrado de outliers

Esta función está basada en la prueba de Tukey, la cual se detalla en el marco teórico (capítulo 1 – outliers). Para poder hacer uso de ésta, se necesita como dato de entrada la base de datos “números”, la cual posee los valores que se quieren analizar; y como data de salida de obtiene el límite mínimo y máximo donde se encuentran los datos no atípicos. Luego, con estos límites, se obtiene la nueva base de datos.

**CÓDIGO EN R**

```
tukeyOutRange=function(numeros, k=1.5){
  RIQ=diff(quantile(numeros,c(0.25,0.75),na.rm = T))
  return(c(quantile(numeros,0.25,na.rm = T)-k*RIQ,
           quantile(numeros,0.75,na.rm = T)+k*RIQ))}
```

### 4.1.2. Distancia entre coordenadas

Este algoritmo tiene como fundamento la distancia más corta entre dos coordenadas en espacios elípticos. Para utilizar esta función se necesitará como input las latitudes "lat1" "lat2" y las longitudes "lon1" "lon2" de ambas coordenadas. Como valor de salida se obtiene la distancia en metros entre ambas coordenadas.

**CÓDIGO EN R**

```
coord2m=function(lat1,lon1,lat2,lon2){
  lat1=lat1*pi/180
  lon1=lon1*pi/180
  lat2=lat2*pi/180
  lon2=lon2*pi/180
  dlat=lat2-lat1
  dlon=lon2-lon1
  a=sin(dlat/2)^2+cos(lat1)*cos(lat2)*sin(dlon/2)^2
  c=2*atan2(sqrt(a),sqrt(1-a))
  return(6373000*c)}
```

### 4.1.3. Aproximación a nodos

El algoritmo de aproximación de nodos sirve para encontrar un nodo, dentro de la base de arcos y nodos del mapa de Lima, que represente al nodo que desea ser aproximado. Como valor de entrada se tiene a "**lats1**" y "**lons1**" que es la base de nodos ya definidos, y "**lats 2**" y "**lons2**", que son los nodos para aproximar.

**CÓDIGO EN R**

```
aprox2Nodes=function(lats1,lons1,lats2,lons2){
  cl <- makeCluster(12)
  registerDoSNOW(cl)
  iterations <- length(lats1)
  pb <- txtProgressBar(max = iterations, style = 3)
  progress <- function(n) setTxtProgressBar(pb, n)
  opts <- list(progress = progress)
  result <- foreach(i = 1:iterations, .combine = c,
                    .options.snow = opts) %dopar%
                    {

return(nodos$ID[which.min(coord2m(lats2,lons2,lats1[i],lons1[i]))])
                    }
  close(pb)
```

```
  stopCluster(cl)
  return(result)}
```

### 4.1.4. Generar variables aleatorias con una distribución

Esta función tendrá como base la densidad de Kernel y utilizará una muestra de los números almacenados previamente, a partir de los cuales se desea generar nuevos números. Se utilizará principalmente para hallar los tiempos simulados.

**CÓDIGO EN R**

```
rkernel=function(samp,N,positiveOnly=F){
  out=sample(samp, N, replace=TRUE) + rnorm(N, 0, bw.nrd0(samp))
  if(positiveOnly){
    while(sum(out<0)>0){
      out[out<0]=sample(samp, sum(out<0), replace=TRUE)+
        rnorm(sum(out<0), 0, bw.nrd0(samp))
    }
  }
  return(out)}
```

### 4.1.5. Generar variables aleatorias con una proporción

Este método utiliza una función acumulada de proporciones y seguirá los siguientes pasos:

- Genera números aleatorios del 0 al 1 (serán los valores en el eje “y”)
- Iguala cada valor generado a la función acumulada y halla el valor de “x”, el cual es el valor hallado.

**PSEUDOCÓDIGO**

```
funcion genNumerosP(xPdf, yPdf, n)
yCdf <- suma acumulada de yPdf
si (yCdf != 0) {
    yCdf <- yCdf/yCdf(Fin)
    numeros <- n numeros aleatorios
    para (i <- 1; i <= numero de numeros; i++) {
          i2 <- encontrar el indice del primer yCdf < numeros(i)
          numeros(i) <- xPdf(i2)
    }
}
Retornar numeros
Fin
```

## 4.2. Depuración de bases

En primer lugar, la información brindada por el CGVBP debe ser tratada debido a que, para su almacenamiento eficiente en la fuente, el formato utilizado y los tipos de variables contienen valores que el software que se utilizará en el desarrollo de la tesis

no reconoce. Existen dos tipos de bases de datos a utilizar: la base del mapa de Lima con nodos y arcos, y la base brindada por el CGBVP.

En ambas bases de datos se eliminarán los valores atípicos ya que no representan a la población analizada y pueden afectar a las posteriores distribuciones halladas. Para poder realizar esta depuración se utilizarán las funciones descritas a continuación.

### 4.2.1. Base del mapa de Lima

Esta base se obtuvo de la página web www.openstreetmap.org, de donde se descargó los nodos y arcos del mapa del Perú en un archivo “osm”, y con esto se pudo obtener la información correspondiente a Lima. Cabe resaltar que no se descargó directamente el mapa de Lima debido a que del programa solo se obtiene bloques menores al tamaño deseado, por lo que es más complicado e inexacto unirlos. Luego de descargar este archivo, se procedió a cortar el mismo en 156 partes por temas de capacidad de procesamiento. Como segundo paso, se utilizó un programa en Visual Basic para poder arreglar los cortes y no omitir información, por ejemplo, no cortar autopistas entre un archivo y otro.

Al arreglar los cortes, se extraen los nodos y arcos de los archivos, y se eliminan los nodos que estén fuera del área a analizar. Luego, se eliminan los arcos que estaban unidos a los nodos eliminados. En esta etapa ya se cuenta con un primer grafo que contiene al grafo de Lima.

Como resultado del paso anterior se obtiene dos bases de datos:

- La primera base cuenta con el nodo inicial y final de cada arco posible, junto con el sentido del arco, el cual puede tener un solo sentido o doble.

| ID nodo 1 | ID nodo 2 | 1w / 2w |
|---|---|---|

- La segunda base está compuesta por todos los nodos con las latitudes y longitudes respectivas.

| ID nodo | Latitud | Longitud |
|---|---|---|

Luego, se ingresan estos datos al software y se procede a eliminar todos los nodos que tengan posiciones fuera de los siguientes rangos para obtener solo el mapa de Lima:

**Latitud**: [-12.5196; -11.5653]

**Longitud:** [-77.2010; -76.6144]

Posteriormente, se eliminan los arcos que estaban conectados a los nodos eliminados, y así obtendríamos una primera versión del grafo de Lima. En la Figura 16.A se observa el mapa de Lima, el cual se desea obtener, y en la Figura 16.B. se observa el mapa con los arcos y nodos que forman Lima.

Para poder obtener el grafo definitivo, se utilizará el algoritmo de búsqueda en anchura (1.3), en donde se generan todos los árboles posibles. Con la ayuda del software, se utilizará el algoritmo mencionado, donde se obtendrá los árboles de mayor a menor cantidad de nodos, de tal manera que se detendrá cuando los nodos restantes a analizar luego de encontrar un árbol sean menores en cantidad al árbol principal.

*Figura 16.A.*

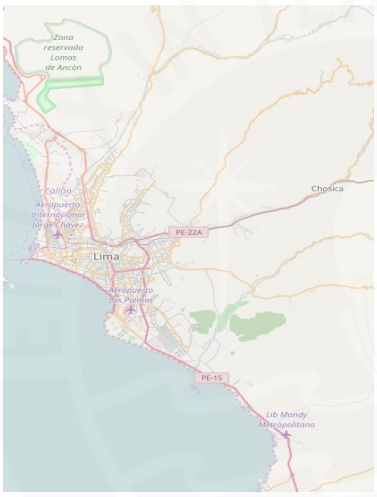



*Figura 16.B.*

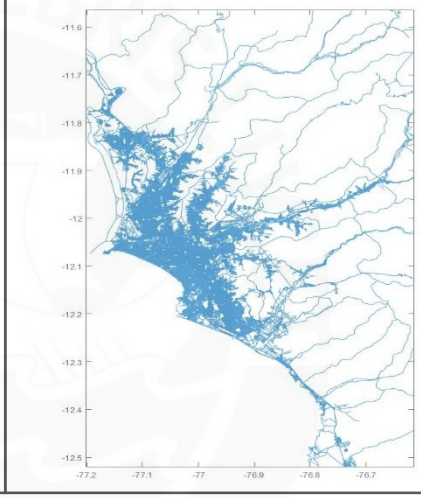



*Figura 16: Representación del grafo de Lima*

*Elaboración propia*

Por ejemplo, en el caso de que se tenga un total 1000 nodos, se encuentran los siguientes árboles:

Árbol 1: 400 nodos
Árbol 2: 150 nodos
Árbol 3: 40 nodos
Árbol 4: 39 nodos

En este caso, para el programa solo quedaría 371 nodos posibles para generar un árbol, cantidad menor a los 400 nodos que se encontraron como árbol mayor, por lo que el programa se detendrá ya que encontró el árbol con mayor cantidad de nodos.

El árbol que se elegirá será el de mayor cantidad de nodos (árbol principal), el cual ya no tendrá nodos inconexos. Finalmente, se duplicarán los arcos que pertenecen

a las calles de doble sentido (1w/2w) y de obtendrá finalmente el grafo de Lima (Figura 17).

*Figura 17: Ejemplo antes y después de la limpieza del grafo*

*Elaboración propia*

El código utilizado para obtener la base de arcos y nodos de Lima se encuentra en el Anexo 2.

### 4.2.2. Base brindada por el CGBVP

El Cuerpo General de Bomberos Voluntarios del Perú compartió la base de datos donde registran la información de cada "Parte" o emergencia nueva a ser atendida. De esta base se extraerá la información útil y será depurada. Los encabezados seleccionados para trabajar son:

*Tabla 19: Estructura de la Base de Partes*

| | |
|---|---|
| **NumParte** | Es el número que se le asigna a un accidente. |
| **Fecha_parte** | Es la fecha donde registra el parte. Se convierte en número. |
| **Codvehi** | Es el código del tipo de vehículo que se manda al accidente. |
| **CodTipoEmer** | Es el código del tipo de emergencia. |
| **X** | Es la longitud de la posición del accidente. |
| **Y** | Es la latitud de la posición del accidente. |
| **Cia** | Es la compañía de la cual se envía el vehículo. |
| **FechaDesp** | Es la fecha en la que se envía la orden de salía del vehículo. Se convierte en número. |
| **FechaSal** | Es la fecha real en la que sale el vehículo de la estación. Se convierte en número. |
| **FechaLleg** | Es la fecha en la que llega el vehículo al lugar de la emergencia. Se convierte en número. |
| **FechaRet** | Es la fecha en la que se retira el vehículo hacia la estación. Se convierte en número. |
| **FechaIng** | Es la fecha en la que llega el vehículo a la estación. Se convierte en número. |
| **Numfalle** | Es el número de fallecidos que originó el accidente. |
| **Numheri** | Es el número de heridos que originó el accidente. |
| **Codubigeo** | Es el código que tiene cada distrito al cual pertenece cada parte. |
| **Tipo_Salida** | Es el código que representa el tipo de salida. |
| **Tipo_vehi** | Es el código que representa el tipo de vehículo. |
| **Es_empalme** | Es el código que representa si el vehículo viene de un empalme. |

*Elaboración propia*

Luego, esta información se importa al software, donde se comienza a realizar la limpieza respectiva. Se procede a extraer cuántos tipos de emergencia existen y cuántos distritos abarcan para obtener una descripción general. Posteriormente, se elimina los duplicados de partes de emergencia para poder extraer los códigos del tipo de emergencia y los códigos de ubigeo. Por otro lado, para cada registro se hallará los tiempos entre eventos DT1, DT2, DT3, DT4 y DT5, a partir de las fechas de parte, despacho, salida, llegada, retorno e ingreso. El detalle de estos datos se presenta en la Figura 18.

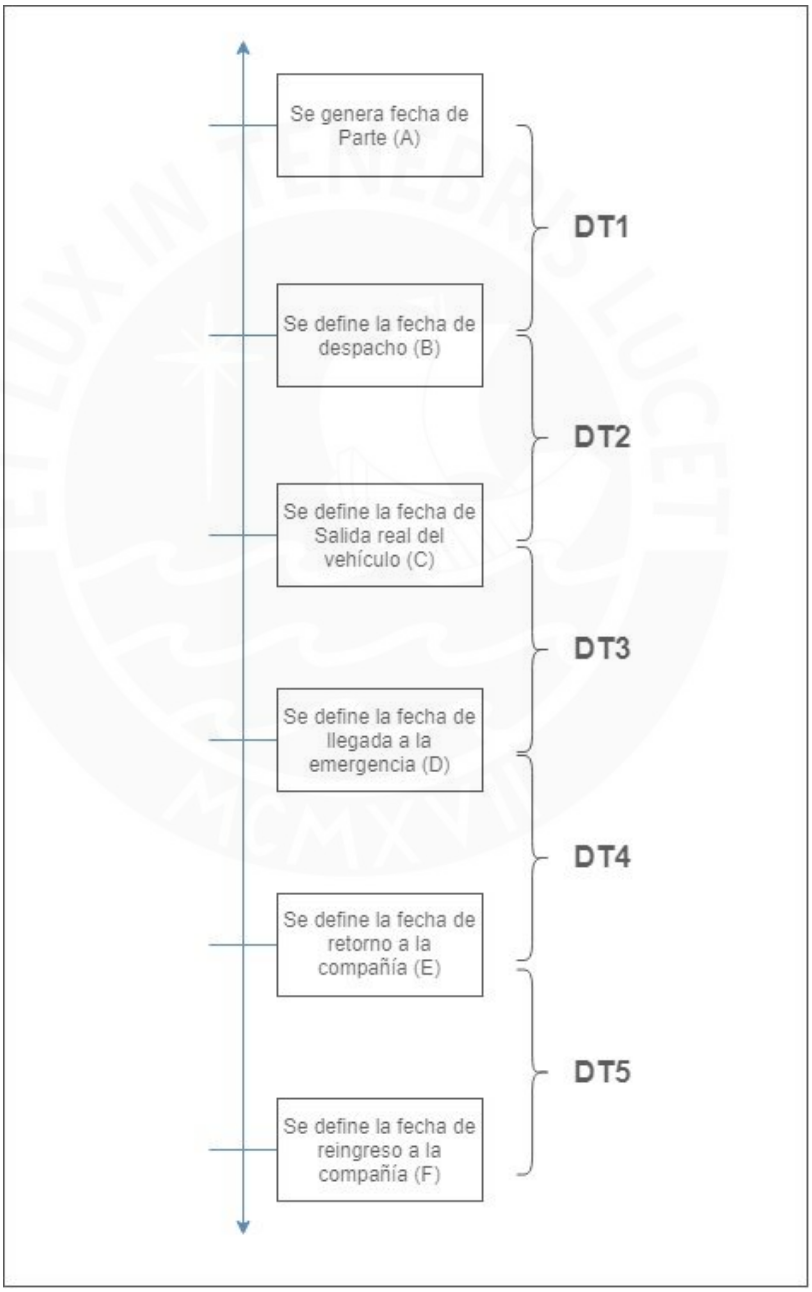



*Figura 18: Deltas de tiempo*

*Elaboración propia*

Se procederá a realizar la limpieza de la base de datos, donde Inicialmente se contaba con 432,341 registros (partes). Se siguen los siguientes pasos:

1. Se eliminarán los registros que tengan latitud y longitud igual a 0, latitud igual a la longitud, latitud vacía y longitud vacía, quedando 191,510 registros.
2. Como segunda depuración se procede a eliminar los tipos de accidentes que son menores a 1 y mayores a 5 (desastres naturales y servicios especiales), esto debido a que cuenta con un porcentaje muy bajo a comparación de las demás emergencias, por otro lado, cuando ocurren este tipo de emergencias se procede a utilizar otro protocolo, donde los sistemas del CGBVP ya no funcionan. Luego de esta depuración se contará con 190,867 registros.
3. Como tercera depuración se eliminarán los partes que no pertenecen a las compañías dentro del área de análisis (Lima) y otras compañías que no existan o no se cuenten con información, obteniendo 190,769 registros.
4. Como cuarta depuración se eliminarán los registros cuyos deltas de tiempo (dt1, dt2, dt3, dt4 y dt5) sean menores que 0, quedando con 145,263 registros.
5. Como quinta depuración se eliminarán los tipos de salida para las ayudas diferentes a la salida normal, quedando con 127,510 registros.
6. Como sexta depuración se eliminarán los registros que vienen de empalmes de accidentes, quedando con 111,882 registros.
7. Como séptima depuración se eliminarán los outliers de velocidad dt3 (velocidad de ida al accidente) utilizando la función borrado de outliers (4.1.1), quedando con 104,558 registros.
8. Como octava depuración se eliminarán los outliers de velocidad dt5 (velocidad de regreso a la estación) utilizando la función borrado de outliers (4.1.1), quedando con 91,483 registros.

El código de depuración utilizado es el siguiente:

**CÓDIGO EN R**

```
#limpieza clase de variable
      partes1[,fecha_parte:=as.POSIXct('1900-01-
      01')%m+%seconds(partes1$fecha_parte*24*60*60)]
      partes1[,fechaDesp:=as.POSIXct('1900-01-
      01')%m+%seconds(partes1$fechaDesp*24*60*60)]
      partes1[,FechaSal:=as.POSIXct('1900-01-
      01')%m+%seconds(partes1$FechaSal*24*60*60)]
      partes1[,FechaLleg:=as.POSIXct('1900-01-
      01')%m+%seconds(partes1$FechaLleg*24*60*60)]
      partes1[,FechaRet:=as.POSIXct('1900-01-
      01')%m+%seconds(partes1$FechaRet*24*60*60)]
      partes1[,FechaIng:=as.POSIXct('1900-01-
```

```
01')%m+%seconds(partes1$FechaIng*24*60*60)]
partes1[,x:=as.numeric(x)]
partes1[,y:=as.numeric(y)]
#limpieza de posicion
elim<-with(partes1,c(elim,which(x==0 | x==y | y==0 | is.na(x)
| is.na(y))))
print(nrow(partes1)-length(unique(elim)))
#limpieza tipo de accidente
elim<-with(partes1,c(elim,which((CodTipoEmer%/%1000000)>5)))
print(nrow(partes1)-length(unique(elim)))
#limpieza de rango lima
elim<-with(partes1,c(elim,
                     which(!(x>-77.2010 & x<(-76.6144) & y>-
12.5196 & y<(-11.5653)))))
print(nrow(partes1)-length(unique(elim)))
#limpieza por cia
elim<-with(partes1,c(elim,which(!cia%in%cias$ID)))
print(nrow(partes1)-length(unique(elim)))
#deltas de tiempo
partes1[,dt1:=interval(fecha_parte,fechaDesp)/duration(1,'seco
nds')]
partes1[,dt2:=interval(fechaDesp,FechaSal)/duration(1,'seconds
')]
partes1[,dt3:=interval(FechaSal,FechaLleg)/duration(1,'seconds
')]
partes1[,dt4:=interval(FechaLleg,FechaRet)/duration(1,'seconds
')]
partes1[,dt5:=interval(FechaRet,FechaIng)/duration(1,'seconds'
)]
elim<-
with(partes1,c(elim,which(!(dt1>=0&dt2>=0&dt3>=0&dt4>=0&dt5>=0
))))
print(nrow(partes1)-length(unique(elim)))
#tipo de salida
elim<-with(partes1,c(elim,which(codigo_clase_salida!=3)))
print(nrow(partes1)-length(unique(elim)))
#empalmes
partes1[2:.N,clase_salida_ant:=partes1$codigo_clase_salida[1:n
row(partes1)-1]]
partes1[2:.N,cod_vehi_ant:=partes1$codvehi[1:nrow(partes1)-1]]
elim<-with(partes1,c(elim,which(clase_salida_ant==2 &
cod_vehi_ant==codvehi)))
partes1[,clase_salida_ant:=NULL]
partes1[,cod_vehi_ant:=NULL]
print(nrow(partes1)-length(unique(elim)))
#Outliers de velocidad
partes1[,x_cia:=cias[match(cia,ID),lon]]
partes1[,y_cia:=cias[match(cia,ID),lat]]
partes1[,dist:=coord2m(y,x,y_cia,x_cia)]
partes1[,vel_dt3:=dist/dt3]
partes1[,vel_dt5:=dist/dt5]
rango1=tukeyOutRange(partes1$vel_dt3[-elim])
rango2=tukeyOutRange(partes1$vel_dt5[-elim])
elim=c(elim,which(!between(partes1$vel_dt3,rango1[1],rango1[2]
```

```
))))
elim=c(elim,which(!between(partes1$vel_dt5,rango2[1],rango2[2]
))))
print(nrow(partes1)-length(unique(elim)))
elim<-unique(elim)
rm(rango1,rango2)
#eliminar
partes1=partes1[!elim,]
partes1=partes1[!is.na(fecha_parte),]
partes1=partes1[!is.na(fechaDesp),]
partes1=partes1[!is.na(FechaSal),]
partes1=partes1[!is.na(FechaLleg),]
partes1=partes1[!is.na(FechaRet),]
partes1=partes1[!is.na(FechaIng),]
rm(elim)
```

Para poder hallar la proporción de los tipos de accidentes se procederá a quitar partes repetidos, quedando con 72,628 registros con la siguiente distribución:

*Tabla 20: Cantidad relativa por tipo de emergencia después del depurado*

| Tipo | Nombre | Cantidad | % |
|---|---|---|---|
| 1 | Incendio | 7,867 | 10,83% |
| 2 | Materiales Peligrosos | 4,324 | 5,95% |
| 3 | Emergencia médica | 50,222 | 68,15% |
| 4 | Rescates | 1,664 | 2,29% |
| 5 | Accidente vehicular | 8,551 | 11,78% |
| 6 | Desastres naturales | 0 | 0% |
| 7 | Servicios Especiales | 0 | 0% |
| | **Total General** | **72,628** | **100%** |

*Elaboración propia*

Para fines prácticos se ha decidido contar con grupos horarios donde cada uno tendrá 4 horas de duración, y en total se obtendrá 42 grupos horarios comenzando desde las 0:00 horas de lunes. Además, se clasificará la información por tipo de accidente (cinco tipos de accidentes) y grupo horario, a lo cual se le denominará bloque horario, quedando inicialmente con 42x5 bloques.

## 4.3. Tratamiento de información

Para cada grupo de información se obtendrá la distribución que mejor ajuste a cada grupo de datos. Se utilizará la totalidad de datos de la población a analizar, esto debido a que se cuenta con la capacidad suficiente de procesamiento. En caso contrario, se propone realizar un muestreo y utilizar las herramientas necesarias para hallar un tamaño de muestra significativo.

### 4.3.1. Tratamiento de información de partes

Para el posterior análisis de la distribución geográfica – temporal es necesario que las emergencias sean separadas por Tipo de accidente y por bloque horario dado que los recursos asignados dependen de esta clasificación.

**Clasificación por tipo, bloque horario**

En este paso se eliminarán los partes de emergencias duplicados para poder clasificarlos por tipo y dependiendo de la hora en que ha sido registrada, se clasificará por bloque horario, obteniendo 42x5 grupos.

**CÓDIGO EN R**

```
#Creación de bloque horario----
      partes1[,bloq_hor:=
                paste('b',
                      floor(interval(as.POSIXlt.character('2013-08-
      05 00:00:00'),fecha_parte)/
                                duration(4,'hours'))%%42+1,sep='_')]
      partes1<-partes1[order(partes1$fecha_parte,decreasing =
      FALSE),]
      partes1[,TipEmergencia:=as.factor(paste('t',CodTipoEmer%/%1000
      000,sep='_'))]
      freq_bloq_tipo<-partes1[,.N,by=.(bloq_hor,TipEmergencia)]
      ggplot(freq_bloq_tipo,aes(x =
      as.numeric(str_sub(bloq_hor,3,nchar(bloq_hor))),
                                y =
      N,colour=TipEmergencia))+geom_line(lwd=2)
```

Los tipos de emergencia con los que trabajaremos son los que se muestran en la Figura 19, donde el eje “x” comprende al bloque horario mencionado líneas arriba y el eje “y” el número de emergencias válidas.

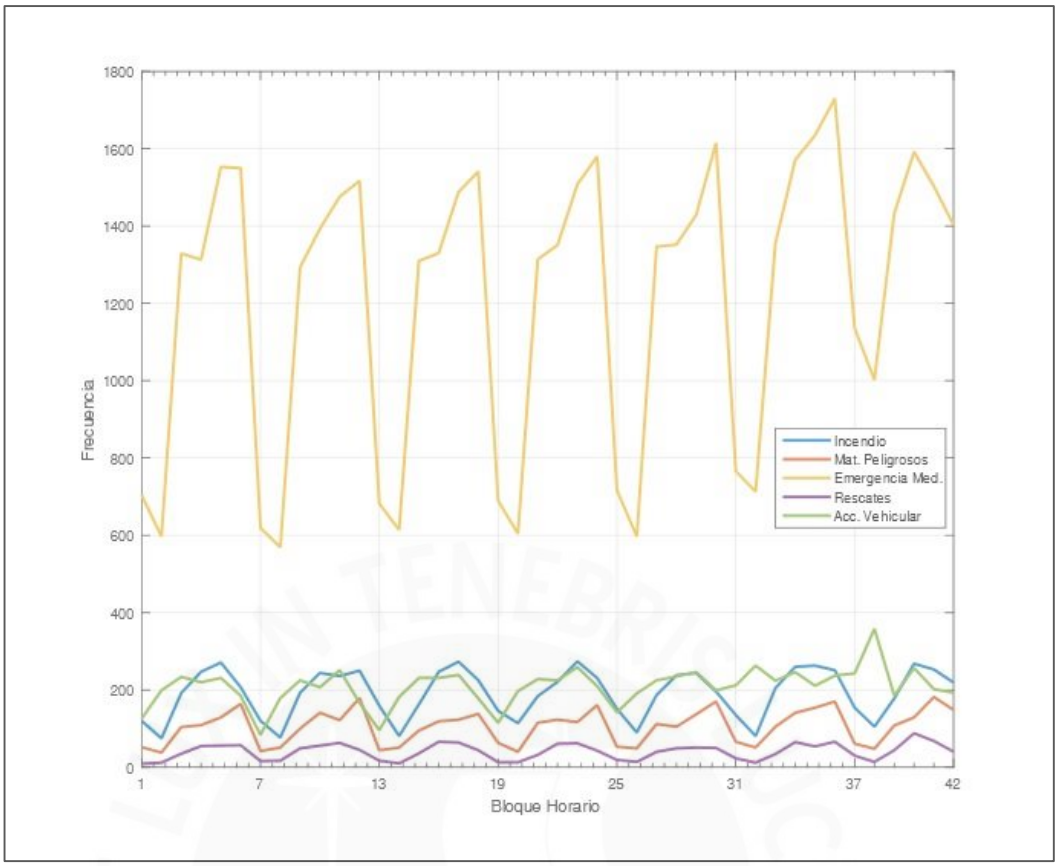



*Figura 19: Tipos de emergencia por bloque horario*

*Elaboración propia*

## Distribución espacial por tipo de emergencia

Al obtener los grupos, se hallará la distribución espacial de las emergencias y se almacenarán en "**spacial_dens**", utilizando el siguiente código:

**CÓDIGO EN R**

```
#Distribución espacial por tipo de emergencias----
      spacial_dens=vector(mode = "list", length =
      length(unique(partes1$TipEmergencia)))
      names(spacial_dens)=sort(unique(partes1$TipEmergencia))
      nx=round(coord2m(mean(partes1$y),min(partes1$x),mean(partes1$y
      ),max(partes1$x))/50)
      ny=round(coord2m(min(partes1$y),mean(partes1$x),max(partes1$y)
      ,mean(partes1$x))/50)
      pb=txtProgressBar(min = 0,max = 42*5,style = 3)
      for (tipo_emer_i in sort(unique(partes1$TipEmergencia))) {
        spacial_dens[[tipo_emer_i]]=vector(mode = "list", length =
      length(unique(partes1$bloq_hor)))

      names(spacial_dens[[tipo_emer_i]])=sort(unique(partes1$bloq_ho
      r))
        for (bloq_hor_i in sort(unique(partes1$bloq_hor))) {
          x=partes1[TipEmergencia==tipo_emer_i &
```

```
bloq_hor==bloq_hor_i,x]
    y=partes1[TipEmergencia==tipo_emer_i &
bloq_hor==bloq_hor_i,y]
    spacial_dens[[tipo_emer_i]][[bloq_hor_i]]=kde2d(x = x,
                                                    y = y,
                                                    n =
c(nx,ny),
                                                    lims =
c(min(partes1$x),max(partes1$x),min(partes1$y),max(partes1$y))
)
    setTxtProgressBar(pb,pb$getVal()+1)
  }
}
close(pb)
rm(x,y,pb,nx,ny,tipo_emer_i,bloq_hor_i)
# image(spacial_dens[['t_3']][['b_12']])
#
df_dens=as.data.table(expand.grid(x=spacial_dens[[1]][[1]]$x,y
=spacial_dens[[1]][[1]]$y))
# df_dens[,z:=as.numeric(spacial_dens[[1]][[1]]$z)]
# ggplot(df_dens,aes(x = x,y = y,fill =
z))+geom_tile()+coord_fixed(ratio = 1)
# rm(df_dens)
```

En la Figura 20 se muestra la distribución para la emergencia tipo 3 (emergencia médica) y bloque horario 5 (lunes de 4p.m. a 8p.m.), donde se observa que en la zona de color amarillo cuenta con mayor concentración de emergencias.

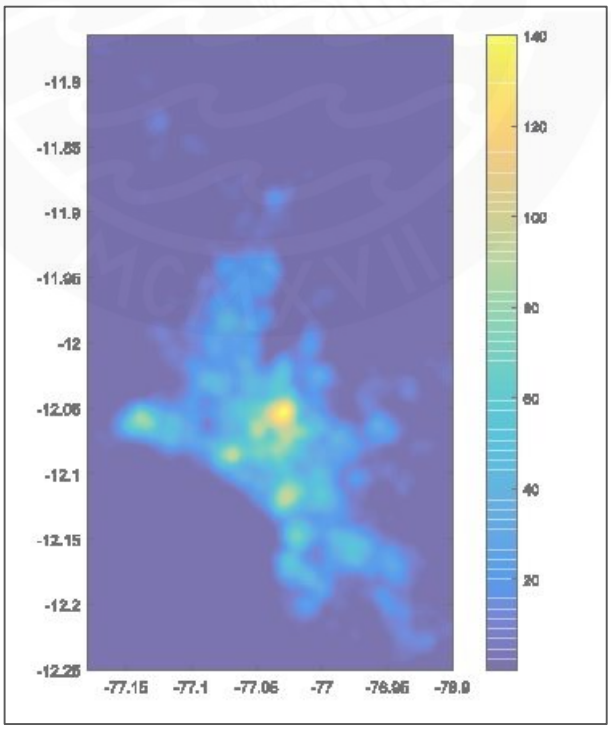



*Figura 20: Distribución de la emergencia tipo 3 en el mapa de Lima*

*Elaboración propia*

**Tiempo entre emergencias por tipo**

Se realizaron análisis previos y se comprobó que el tiempo entre emergencias o accidentes se ajusta a una distribución exponencial. En la Figura 21 se muestra el histograma de una muestra de datos de tiempos entre accidentes de un tipo de emergencia en un bloque horario y la función de distribución exponencial que más se ajusta.

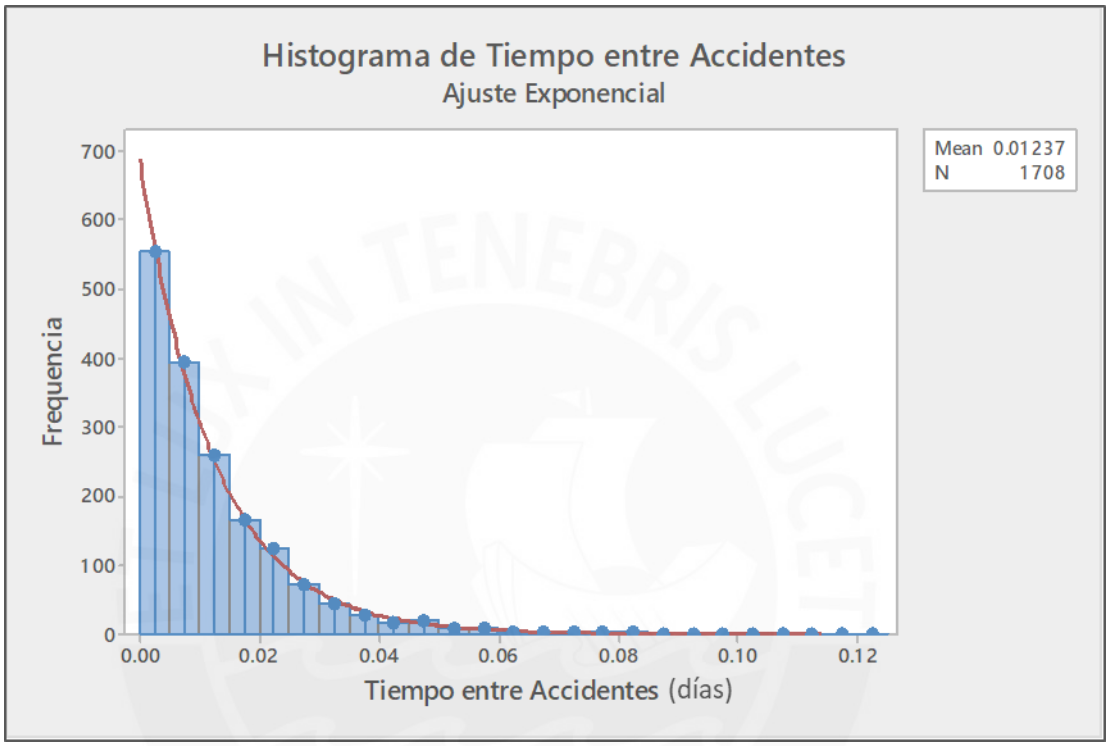



*Figura 21: Histograma de tiempo entre accidentes*
*Elaboración propia*

Además, para validar este supuesto, se realizó la prueba de bondad de ajuste chi-cuadrado y se obtuvo que no se puede rechazar la hipótesis nula, donde se afirma que la muestra proviene de una distribución exponencial.

**CÓDIGO EN MATLAB**

```
Prueba de hipótesis chi-cuadrado

>> pd = fitdist(ans,'Exponential');
>> h = chi2gof(ans,'CDF',pd)
h = 0
```

Debido a que se comprobó que el tiempo entre accidente sigue una distribución exponencial, se procederá a extraer los tiempos promedio entre emergencias por cada tipo y por bloque horario, ya que es el único parámetro necesario para la

distribución exponencial y según el método de máxima verosimilitud, es un estadístico suficiente e insesgado, y se almacenará la información en la matriz "**lambda**" (5 tipos de emergencia x 42 bloques horarios).

**CÓDIGO EN R**

```
#Distribución tiempo entre emergencias----
ajuste_exp=as.data.table(expand.grid(bloq_hor=sort(unique(partes1$b
loq_hor)),tipo_emer=sort(unique(partes1$TipEmergencia))))
ajuste_exp[,lambda:=NA_real_]
pb=txtProgressBar(min = 0,max = nrow(ajuste_exp),style = 3)
for (tipo_emer_i in unique(ajuste_exp$tipo_emer)) {

dt_temp=partes1[TipEmergencia==tipo_emer_i,][!duplicated(numParte),
]
  dt_temp[1:(.N-
1),t_entre_emer:=interval(fecha_parte,dt_temp[2:.N,fecha_parte])/du
ration(1,'seconds')]
  for (bloq_hor_i in unique(ajuste_exp$bloq_hor)) {
    tiempo_entre_emer=dt_temp[!is.na(t_entre_emer) &
bloq_hor==bloq_hor_i,t_entre_emer]
    fit=fitdistr(tiempo_entre_emer,densfun = 'exponential') #ajuste
exponencial
    # cat(tipo_emer_i,',',bloq_hor_i,':\n')
    # cat(1/6*exp(-CramerVonMisesTwoSamples(S1 =
tiempo_entre_emer,S2 = rexp(1000,fit$estimate))),'\n')
    # hist_temp=hist(tiempo_entre_emer,probability = T)
    # curve(dexp(x,fit$estimate),add=T,col=2)
    # readline(prompt="Press [enter] to continue")
    ajuste_exp[bloq_hor==bloq_hor_i &
tipo_emer==tipo_emer_i,lambda:=1/fit$estimate]
    setTxtProgressBar(pb,pb$getVal()+1)
  }
}
close(pb)
rm(fit,bloq_hor_i,tipo_emer_i,tiempo_entre_emer,pb,dt_temp)
```

**Depuración de DT1**

Para tener en consideración las posibles fallas de los vehículos y la repercusión que genera la falta de disponibilidad de estos recursos, se realizará el siguiente análisis a la diferencia de tiempo entre la fecha de despacho y la fecha en la que se genera el parte. En este caso, se desea contabilizar el tiempo desde que el vehículo está realmente disponible luego del registro del parte de la emergencia hasta el momento del despacho, por lo que se hace un seguimiento a cada vehículo por separado. En la Figura 22 se diagrama los dos casos del tiempo DT1.

$$DT1 = \begin{cases} T3 - T1, & T2 < T1 \\ T3 - T2, & T2 \geq T1 \end{cases}$$

$Donde$:

$T1$: *Hora de Registro de Parte*

$T2$: *Hora de Vehículo Disponible*

$T3$: *Hora de Despacho del Vehículo*

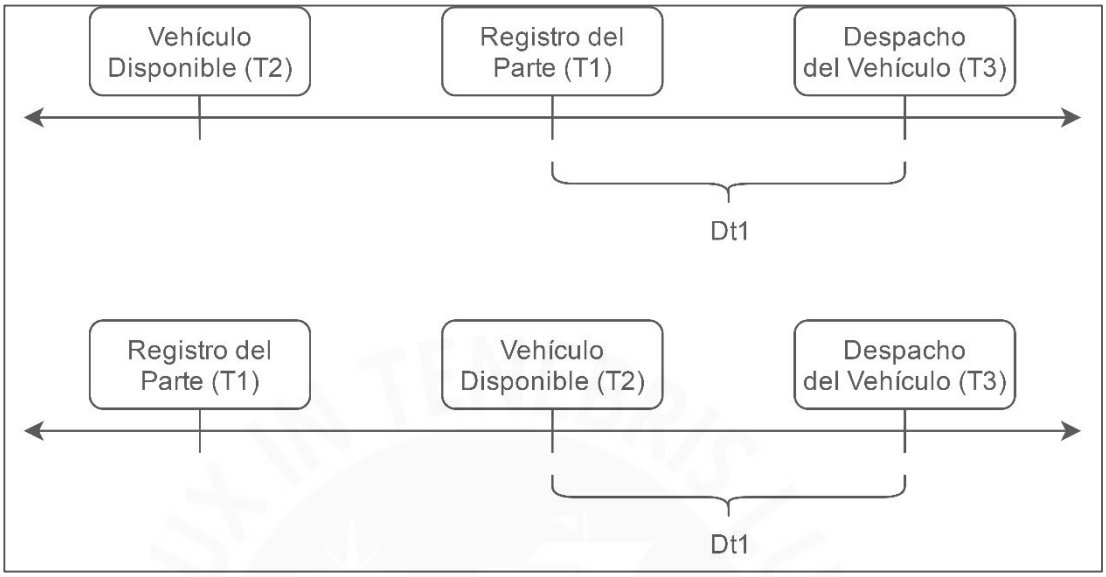



*Figura 22: Representación de los casos de DT1*

*Elaboración propia*

Las matrices de respuesta (5 tipos de emergencias x 13 tipos de vehículos) se almacenan en "**dt1_2_dens**".

**CÓDIGO EN R**

```
partes1[,indice_vehi:=1:.N,by=codvehi]
partes1[,FechaIng_ant:=partes1$FechaIng[match(paste(codvehi,indice_
vehi,sep='-'),

paste(partes1$codvehi,partes1$indice_vehi+1,sep='-'))]]
partes1[,dt1_2:=interval(pmax(fecha_parte,FechaIng_ant,na.rm=T),fec
haDesp)/duration(1,'seconds')]
dt1_2_dens=vector(mode = "list", length =
length(unique(partes1$TipEmergencia)))
names(dt1_2_dens)=sort(unique(partes1$TipEmergencia))
pb=txtProgressBar(min = 0,max = 13*5,style = 3)
for (tipo_emer_i in names(dt1_2_dens)) {
  dt1_2_dens[[tipo_emer_i]]=vector(mode = "list", length =
length(unique(partes1$tip_vehi)))
  names(dt1_2_dens[[tipo_emer_i]])=sort(unique(partes1$tip_vehi))
  for (tip_vehi_i in names(dt1_2_dens[[tipo_emer_i]])) {
    x=partes1[TipEmergencia==tipo_emer_i & tip_vehi==tip_vehi_i &
dt1_2>=0,dt1_2]
    if(length(x)<=1){
      x=partes1[tip_vehi==tip_vehi_i & dt1_2>0,dt1_2]
```

```
    }
    rng=tukeyOutRange(x)
    x=x[between(x,rng[1],rng[2])]
    dt1_2_dens[[tipo_emer_i]][[tip_vehi_i]]=x#density(x)
    setTxtProgressBar(pb,pb$getVal()+1)
  }
}
close(pb)
rm(x,pb,tipo_emer_i,tip_vehi_i,rng)
```

**Depuración de DT2**

Los datos extraídos de DT2 (diferencia del tiempo entre despacho del vehículo y salida de la estación) son depurados y separados por tipo de vehículo, y la información se almacena en 13 matrices unidimensionales las cuales a su vez son almacenadas en "**dt2_dens**".

**CÓDIGO EN R**

```
#Distribución para dt2----
dt2_dens=vector(mode = "list", length =
length(unique(partes1$tip_vehi)))
names(dt2_dens)=sort(unique(partes1$tip_vehi))
pb=txtProgressBar(min = 0,max = length(dt2_dens),style = 3)
for (tip_vehi_i in names(dt2_dens)) {
  x=partes1[tip_vehi==tip_vehi_i,dt2]
  rng=tukeyOutRange(x)
  x=x[between(x,rng[1],rng[2])]
  dt2_dens[[tip_vehi_i]]=x#density(x)
  setTxtProgressBar(pb,pb$getVal()+1)
}
close(pb)
rm(pb,tip_vehi_i,rng)
```

Cabe resaltar que lo hallado no depende del tipo de emergencia ni del bloque horario, como se muestra en la Figura 23, en la que DT2 ha sido clasificada por tipo de emergencia.

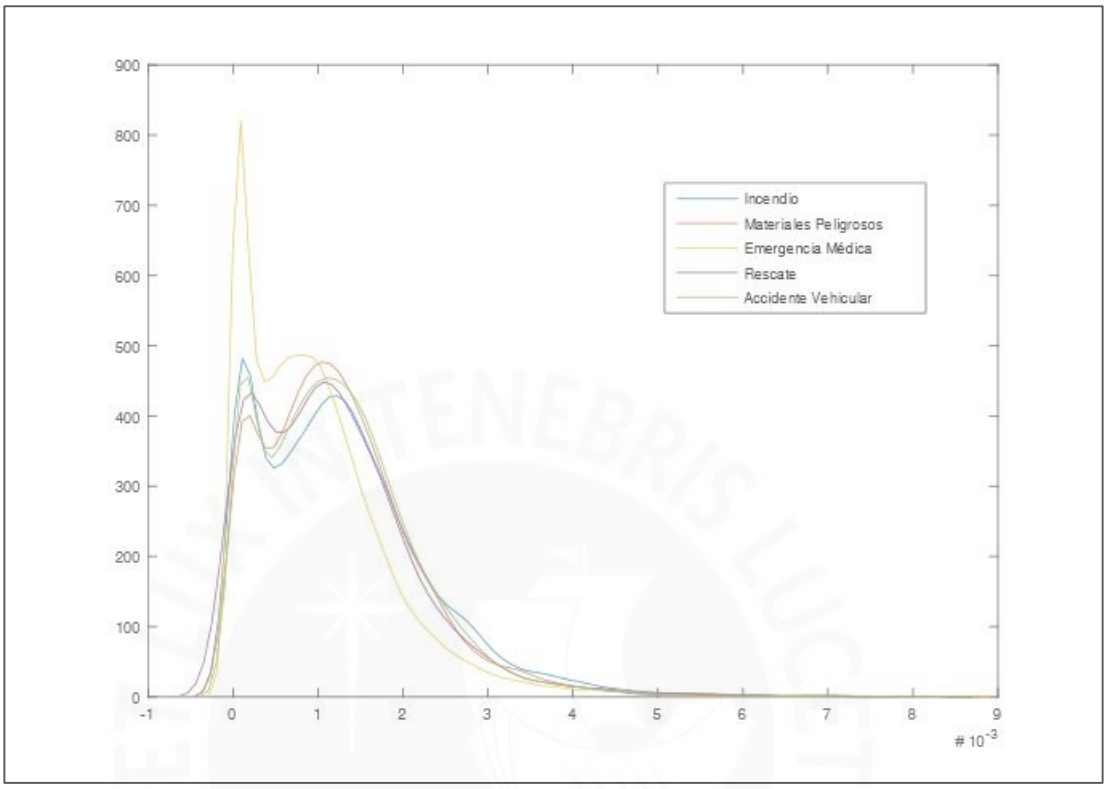



*Figura 23: Distribución de DT2 según tipo de emergencia*

*Elaboración propia*

## Depuración de DT4

Los datos extraídos de DT4 (delta de tiempo entre llegada del vehículo y retirada de la emergencia) son depurados y separados por tipo de vehículo y por tipo de emergencia, esto debido a que la utilización de los vehículos depende del tipo de emergencia. La información se almacena en 5x13 matrices unidimensionales las cuales a su vez son almacenadas en "**dt4_dens**".

**CÓDIGO EN R**

```
#Distribución para dt4----
dt4_dens=vector(mode = "list", length =
length(unique(partes1$TipEmergencia)))
names(dt4_dens)=sort(unique(partes1$TipEmergencia))
pb=txtProgressBar(min = 0,max = 13*5,style = 3)
for (tipo_emer_i in names(dt4_dens)) {
  dt4_dens[[tipo_emer_i]]=vector(mode = "list", length =
length(unique(partes1$tip_vehi)))
  names(dt4_dens[[tipo_emer_i]])=sort(unique(partes1$tip_vehi))
```

```
  for (tip_vehi_i in names(dt4_dens[[tipo_emer_i]])) {
    x=partes1[TipEmergencia==tipo_emer_i &
tip_vehi==tip_vehi_i,dt4]
    if(length(x)<=1){
      x=partes1[tip_vehi==tip_vehi_i,dt4]
    }
    rng=tukeyOutRange(x)
    x=x[between(x,rng[1],rng[2])]
    dt4_dens[[tipo_emer_i]][[tip_vehi_i]]=x#density(x)
    setTxtProgressBar(pb,pb$getVal()+1)
  }
}
close(pb)
rm(x,pb,tipo_emer_i,tip_vehi_i,rng)
```

En la Figura 24 se observa que las distribuciones de tiempo DT4 tienen una marcada variación por tipo de emergencia.

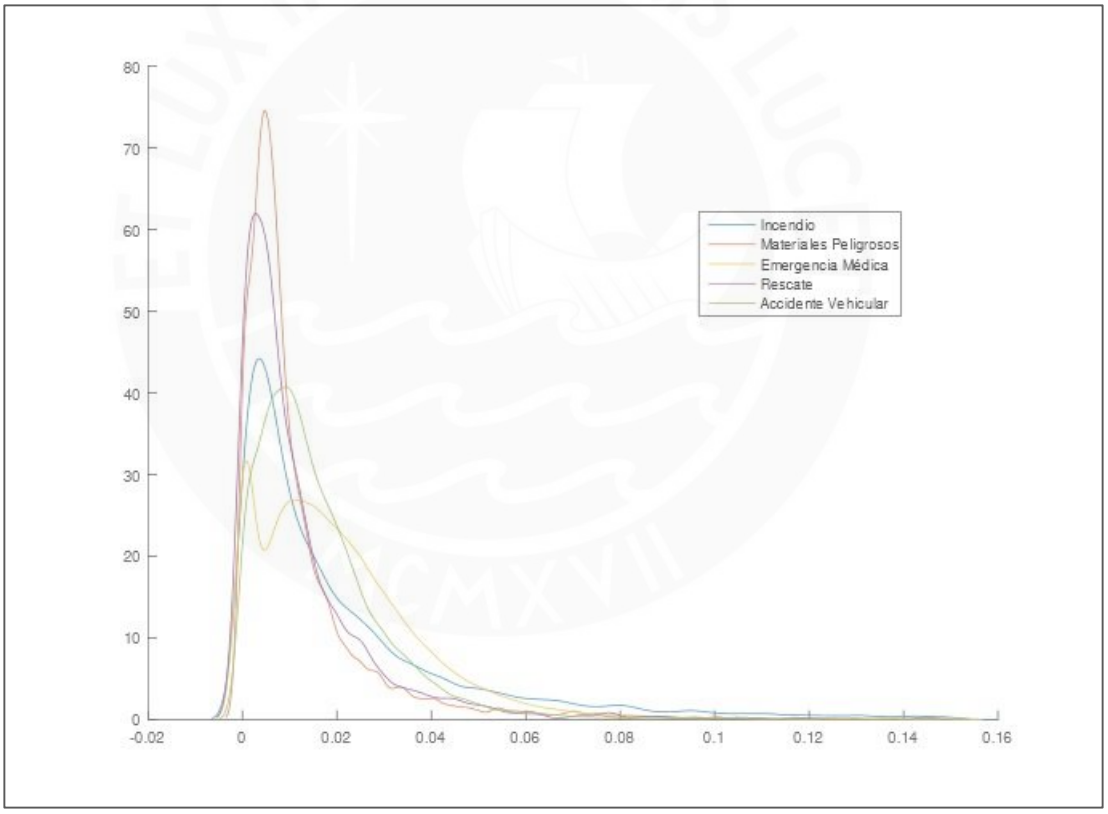



*Figura 24: Distribución de DT4 según tipo de emergencia*

*Elaboración propia*

## Estado inicial de los recursos

Debido a que no se cuenta con una base de datos donde se indique los vehículos disponibles, esta información se aproximará de la base de partes. Lo que se plantea en el código que se encuentra líneas abajo, es que se halle el promedio de las diferencias de tiempo entre cada uso de un mismo vehículo y su desviación estándar,

y luego si la diferencia del tiempo actual (el último tiempo registrado en la base de partes) y el último tiempo de aparición del vehículo es mayor que el promedio hallado anteriormente más “k” veces la desviación estándar, entonces se considera que el vehículo ya no se encuentra en circulación. Esta información se almacenará en **vehi_dt**.

**CÓDIGO EN R**

```
#Estado Inicial vehículos----
      vehi_dt=partes1[,.(ult_fecha=max(FechaIng,na.rm=T),cia=last(ci
      a),tip_vehi=last(tip_vehi)),by=codvehi]
      vehi_dt[,t_desde_uso:=interval(ult_fecha,max(ult_fecha))/durat
      ion(1,'seconds')]
      vehi_dt[,elim:=t_desde_uso>.(min(t_desde_uso)+ifelse(is.na(sd(
      t_desde_uso)),0,sd(t_desde_uso))*0.6),by=tip_vehi]
      vehi_dt=vehi_dt[!(elim),]
      vehi_dt[,.N,by=c('cia','tip_vehi')]
      disp_vehi=dcast(data =
      vehi_dt[,.N,by=c('cia','tip_vehi')],formula =
      cia~tip_vehi,fun.aggregate = sum,fill = 0,value.var = 'N')
```

La cantidad de recursos iniciales hallados se encuentra en el Anexo 3.

### 4.3.2. Tratamiento de información de compañías

En las bases mencionadas anteriormente se encuentra el encabezado **Codidenest**, el cual es el código que pertenece a cada compañía. Cada una cuenta con una latitud y longitud conocida. Para fines prácticos, se aproximará cada una de estas posiciones a nodos que ya existan en el grafo de Lima hallado anteriormente utilizando la función **coord2m** (4.1.2).

**CÓDIGO EN R**

```
for(cia in 1:nrow(base_cias)){
  d=coord2m(nodos$lat,nodos$lon,base_cias$lat[cia],base_cias$lon[ci
a])
  base_cias[cia,ID_nodo:=nodos[which.min(d),ID]]
  base_cias[cia,dist_nodo:=min(d)]
}
rm(d,cia)
fwrite(base_cias,'ciasLima.dat')
```

### 4.3.3. Tratamiento de información de vehículos

Previamente se asignará una matriz de relación entre cada vehículo y la comandancia a la que pertenece utilizando la base de vehículos.

**Proporción de vehículos por tipo y cantidad por emergencia**

Como siguiente paso, se hallarán las proporciones del número y el tipo de vehículos a enviar a una emergencia, según el bloque horario y el tipo de emergencias. Para hallar estos datos, se usará el algoritmo de distribución del número de vehículos y el algoritmo de distribución del tipo de vehículos, desarrollados líneas abajo, y se obtendrá "**nVehi**" (5 tipos de emergencias x 42 bloques horarios x 88 vehículos máximo por emergencia) y "**probVehi**", la cual es una matriz de 3 dimensiones (5 tipos de emergencias x 42 bloques horarios x 13 tipos de vehículo).

**CÓDIGO EN R**

```
#Distribución de Número de vehículos----
    nVehi=vector(mode = 'list',length = 5)
    for (te in 1:5) {
      nVehi[[te]]=vector(mode = 'list',length = 42)
      for (bh in 1:42) {
        tabla=table(partes1[TipEmergencia==paste('t',te,sep='_') &
    bloq_hor==paste('b',bh,sep='_'),.N,by=numParte]$N)

    nVehi[[te]][[bh]]=as.numeric(tabla[match(1:88,names(tabla))]/sum
    (tabla))
        nVehi[[te]][[bh]][is.na(nVehi[[te]][[bh]])]=0
      }
    }
    rm(te,bh,tabla)

#Distribución de Tipo de vehículos----
    probVehi=vector(mode = 'list',length = 5)
    for (te in 1:5) {
      probVehi[[te]]=vector(mode = 'list',length = 42)
      for (bh in 1:42) {

    probVehi[[te]][[bh]]=partes1[TipEmergencia==paste('t',te,sep='_'
    ) &
    bloq_hor==paste('b',bh,sep='_'),.N,by=tip_vehi][match(names(disp
    _vehi)[-1],tip_vehi),N/sum(N,na.rm=T)]
        probVehi[[te]][[bh]][is.na(probVehi[[te]][[bh]])]=0
      }
    }
    rm(te,bh)
```

## 4.4. Preprocesamiento de información de distancias

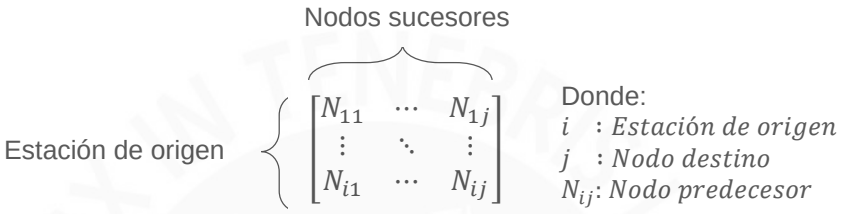
Para aumentar la velocidad de procesamiento del simulador y hacerlo más eficiente se debe compensar la capacidad de procesamiento con espacio de almacenamiento. Varias de las variables a utilizar durante la simulación serán preprocesadas en lotes para aumentar su eficiencia y evitar reprocesar información que ya se utilizó anteriormente.

### 4.4.1. Almacenamiento de matriz de distancias

Luego de aproximar cada comandancia a un nodo ya existente, se procederá a crear las siguientes matrices:

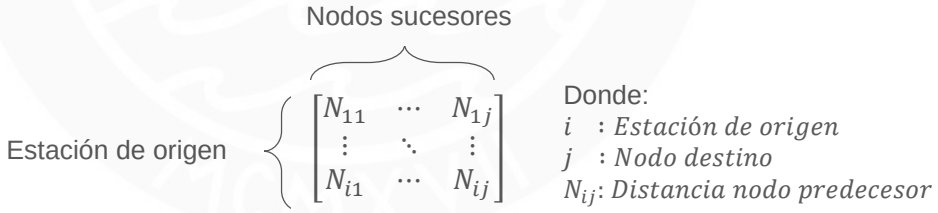
- Matriz de comandancia a nodo predecesor

En esta matriz se tendrá los datos de los nodos predecesores y sucesores correspondientes a cada compañía, y tendrá la siguiente forma: las filas indican las compañías, las columnas indican los nodos sucesores y en los cruces los predecesores. La matriz será de la forma:



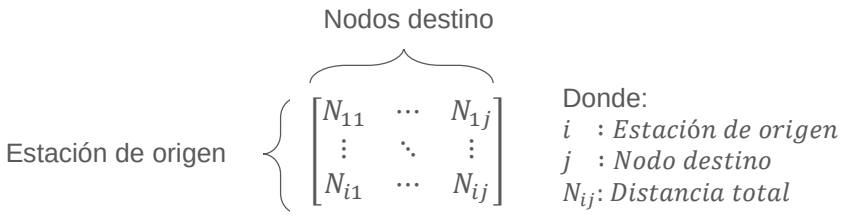

- Matriz de distancias anterior

En esta matriz se tendrá los datos de las distancias de los nodos predecesores a los nodos sucesores por cada compañía siguiendo la ruta más corta, utilizando el método basado en el algoritmo de Dijkstra.




- Matriz de distancia total

En esta matriz se tendrá la distancia total desde cualquier estación a cualquier nodo.




Las matrices se han guardado de esa manera para poder optimizar el espacio en el software. Los códigos utilizados para hallar las matrices de distancia de ida y de

retorno se encuentran en el Anexo 4. Si se desea, se puede reconstruir la ruta desde un nodo a una estación de bomberos utilizando el código del Anexo 5.

## 4.5. Distribuciones de tráfico

Debido a que no se cuenta con información del tráfico en Lima, se procede a utilizar la data histórica de la base de partes, de donde se extrae los tiempos de traslado de cada registro y las distancias calculadas anteriormente para hallar el tráfico promedio según el bloque horario para cada registro y, así, ajustarlo a una distribución de velocidad. Esta información se almacena en "**vel_ida_pdf**" que contiene el tráfico de ida en 42 matrices unidimensionales, y en "**vel_ret_pdf**" que contiene el tráfico de retorno en 42 matrices unidimensionales.

**CÓDIGO EN R:**

```
#velocidad ida y retorno----
      for (cia_ind in 1:nrow(cias)) {

      partes1[cia==cias$ID[cia_ind],distancia_ida:=dist_ida[match(no
      do_id,nodos$ID),cia_ind]]
      }
      for (cia_ind in 1:nrow(cias)) {

      partes1[cia==cias$ID[cia_ind],distancia_ret:=dist_ret[match(no
      do_id,nodos$ID),cia_ind]]
      }
      rm(cia_ind)
      partes1[,vel_ida:=distancia_ida/dt3]
      partes1[,vel_ret:=distancia_ret/dt5]
      partes1[,bloq_hor_ida:=date2BH(FechaSal)]
      partes1[,bloq_hor_ret:=date2BH(FechaRet)]
      vel_ida_pdf=vector(mode = 'list',length = 42)
      vel_ret_pdf=vector(mode = 'list',length = 42)
      for (bloque in 1:42) {

      vel_ida_pdf[[bloque]]=partes1[bloq_hor_ida==paste('b',bloque,s
      ep='_'),vel_ida]

      vel_ret_pdf[[bloque]]=partes1[bloq_hor_ret==paste('b',bloque,s
      ep='_'),vel_ret]
      }
```

En la Figura 25 se muestra la inversa de la velocidad por bloque horario.

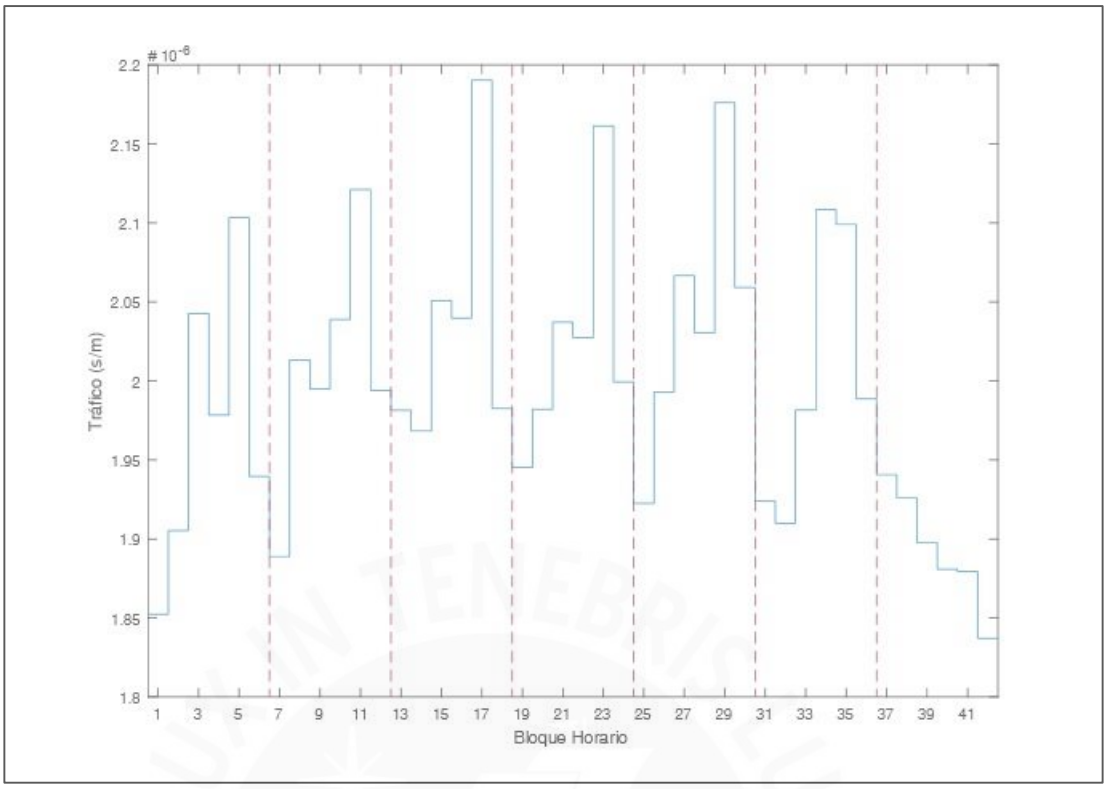



*Figura 25: Tráfico promedio por bloque horario*

*Elaboración propia*

## 4.6. Presimulación de emergencias

De la misma manera que con la información de distancias, es mucho más eficiente realizar la presimulación de las emergencias a utilizar previamente y en lotes. Estos datos serán preasignados y almacenados en una nueva matriz para su posterior uso. Cabe resaltar que se presimularán las variables que no dependan del estado del sistema, hasta cumplir con el tiempo total de simulación de 3 años. El código completo de la presimulación se encuentra en el Anexo 6 y en la Figura 26 se presenta el flujograma propuesta para la simulación.

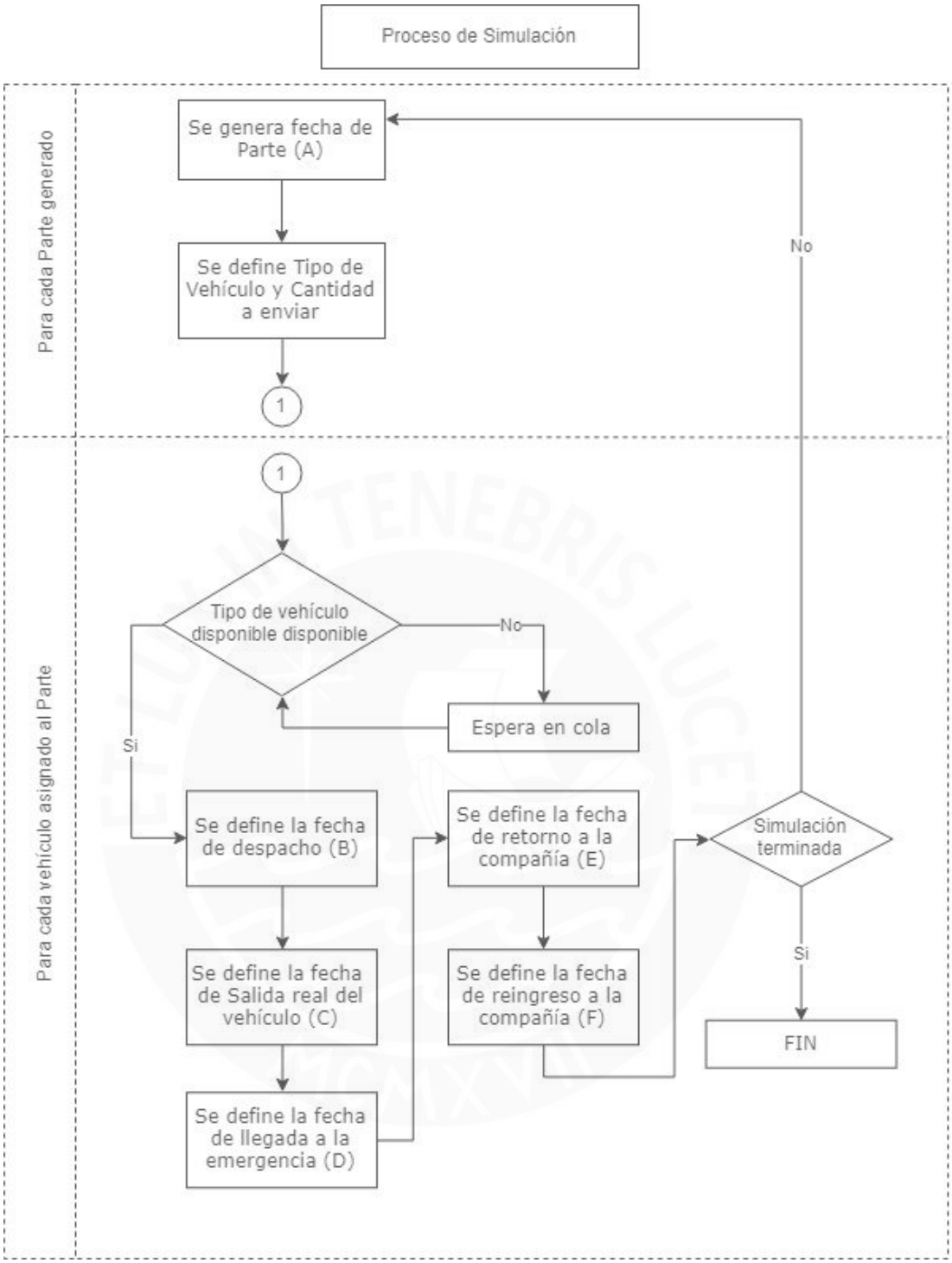



*Figura 26: Flujograma de la simulación*

*Elaboración propia*

### 4.6.1. Presimulación de los tiempos entre emergencias

Con la matriz "**lambda**" hallada anteriormente, se simularán los tiempos entre emergencias, de esta manera, la suma de todos los tiempos debe ser igual al tiempo total a simular.

Cabe resaltar que además de realizar la presimulación por tipo de accidente, se realizará por bloque horario, donde se iniciará en el bloque 1 (tiempo = 0) y se irá recalculando el bloque conforme se acumulen los tiempos.

Finalmente, los tiempos simulados se agrupan y se ordenan de forma ascendente por fecha de registro. A continuación, se muestra una parte del código utilizado:

**CÓDIGO EN R**

```
for (te in 1:5) {
        fecha[[te]]=rexp(n = 1,rate = 1/ajuste_exp[bloq_hor=='b_1' &
      tipo_emer==paste('t',te,sep='_'),lambda])
        pb=txtProgressBar(min = 0,max = 1,style = 3)

        while (last(fecha[[te]])<maxSec) {
          fecha[[te]]=c(fecha[[te]],last(fecha[[te]])+rexp(n =
      1,rate = 1/ajuste_exp[bloq_hor==sec2BH(last(fecha[[te]])) &
      tipo_emer==paste('t',te,sep='_'),lambda]))
          setTxtProgressBar(pb,last(fecha[[te]])/maxSec)
        }
        close(pb) }

#Fecha de la emergencia
      partes_nuevo[,TipEmergencia:=rep(1:5,unlist(lapply(fecha,lengt
      h)))]
      for (tipEmer in 1:5) {

      partes_nuevo[TipEmergencia==tipEmer,fecha_parte:=fecha[[tipEme
      r]]]
      }
      rm(tipEmer)
      partes_nuevo[,bloq_hor:=sec2BH(fecha_parte)]
```

### 4.6.2. Presimulación de las posiciones de las emergencias

Dadas las distribuciones espaciales halladas almacenadas en "**spacial_dens**", se procederá a simular una posición para cada emergencia presimulada en el paso anterior, teniendo en cuenta su tipo y bloque horario. Posteriormente se realizará la aproximación de estos valores a sus nodos más cercanos para poder posicionarlos de manera correcta dentro del grafo utilizando la función "**aprox2Nodes**".

Para hallar la distribución y la generación de nuevos accidentes se utiliza el 0siguiente código.

**CÓDIGO EN R**

```
#Simular posición de emergencias
      set.seed(598453)
      pb=txtProgressBar(max = 5*42,style = 3)
      for (te in 1:5) {
        for (bh in 1:42) {

      distAct=spacial_dens[[paste('t',te,sep='_')]][[paste('b',bh,se
      p='_')]]
          dens=melt(distAct$z)

      indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') &
      partes_nuevo$TipEmergencia==te)
          z_index=sample(x = 1:nrow(dens),size =
      length(indPartes),replace = T,prob = dens$value)
          bw_x=mean(diff(distAct$x))
          bw_y=mean(diff(distAct$y))

      partes_nuevo[indPartes,lon:=distAct$x[dens$Var1[z_index]]+runi
      f(length(indPartes),-bw_x/2,bw_x/2)]

      partes_nuevo[indPartes,lat:=distAct$y[dens$Var2[z_index]]+runi
      f(length(indPartes),-bw_y/2,bw_y/2)]
          setTxtProgressBar(pb,(te-1)*42+bh)
        }
      }
      close(pb)
      rm(pb,te,bh,distAct,dens,indPartes,z_index,bw_x,bw_y)


#Aproximar a nodos (Aprox2nodes)
      cl <- makeCluster(12)
      registerDoSNOW(cl)
      iterations <- nrow(partes_nuevo)
      pb <- txtProgressBar(max = iterations, style = 3)
      progress <- function(n) setTxtProgressBar(pb, n)
      opts <- list(progress = progress)
      result <- foreach(i = 1:iterations, .combine = c,
                        .options.snow = opts) %dopar%
                        {

      return(nodos$ID[which.min(coord2m(nodos$lat,nodos$lon,partes_n
      uevo$lat[i],partes_nuevo$lon[i]))])
                        }
      close(pb)
      stopCluster(cl)
      partes_nuevo[,nodo_id:=result]
      rm(cl,iterations,pb,progress,opts,result)
```

### 4.6.3. Simulación del número de vehículos por emergencia

En este paso se utiliza la matriz "**nVehi**" hallada anteriormente, para generar el número de vehículos que serán utilizados para cada emergencia según su tipo y bloque horario. Adicionalmente, cada emergencia es replicada según el número de vehículos que le corresponde.

**CÓDIGO EN R**

```
#Número de vehiculos
      partes_nuevo[,nVehis:=NA_integer_]
      set.seed(46887)
      pb=txtProgressBar(max = 5*42,style = 3)
      for (te in 1:5) {
        for (bh in 1:42) {

      indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') &
      partes_nuevo$TipEmergencia==te)
          partes_nuevo[indPartes,nVehis:=sample(x = 1:88,size =
      length(indPartes),replace = T,prob = nVehi[[te]][[bh]])]
          setTxtProgressBar(pb,(te-1)*42+bh)
        }
      }
      close(pb)
      rm(pb,te,bh,indPartes)
```

### 4.6.4. Simulación de los tipos de vehículo

Con la matriz de 3 dimensiones "**probVehi**", hallada anteriormente, se genera los tipos de vehículos correspondientes a cada registro simulado.

**CÓDIGO EN R**

```
#Tipos de vehiculos
partes_nuevo=partes_nuevo[rep(1:.N,nVehis),]
partes_nuevo[,tip_vehi:=NA_character_]
set.seed(46887)
pb=txtProgressBar(max = 5*42,style = 3)
for (te in 1:5) {
  for (bh in 1:42) {
    indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') &
partes_nuevo$TipEmergencia==te)
    partes_nuevo[indPartes,tip_vehi:=names(disp_vehi)[sample(x =
1:13,size = length(indPartes),replace = T,prob =
probVehi[[te]][[bh]])+1]]
    setTxtProgressBar(pb,(te-1)*42+bh)
  }
}
close(pb)
rm(pb,te,bh,indPartes)
```

### 4.6.5. Simulación del dt1

En este paso se utilizan las matrices almacenadas en "**dt1_2_dens**" para simular el tiempo entre la fecha de registro y despacho de cada vehículo (DT1) por tipo de emergencia y tipo de vehículo.

**CÓDIGO EN R**

```
#dt1
      set.seed(9541)
      pb=txtProgressBar(max = 5*13,style = 3)
      tipVehiNames=names(disp_vehi)[-1]
      for (te in 1:5) {
        for (tv in 1:13) {
          indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]]
      & partes_nuevo$TipEmergencia==te)

      partes_nuevo[indPartes,dt1:=rkernel(dt1_2_dens[[paste('t',te,s
      ep='_')]][[tipVehiNames[[tv]]]],N = length(indPartes),T)]
          setTxtProgressBar(pb,(te-1)*13+tv)
        }
      }
      close(pb)
      rm(pb,te,tv,indPartes)
```

### 4.6.6. Simulación del dt2

En este paso se utilizan las matrices almacenadas en "**dt2_dens**" para simular el tiempo entre el despacho y salida de cada vehículo (DT2) por tipo de vehículo.

**CÓDIGO EN R**

```
#dt2
      set.seed(4567)
      pb=txtProgressBar(max = 13,style = 3)
      for (tv in 1:13) {
        indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]])

      partes_nuevo[indPartes,dt2:=rkernel(dt2_dens[[tipVehiNames[[tv
      ]]]],N = length(indPartes),T)]
        setTxtProgressBar(pb,tv)
      }
      close(pb)
      rm(pb,tv,indPartes)
```

### 4.6.7. Simulación del dt4

En este paso se utilizan las matrices almacenadas en "**dt4_dens**" para simular el tiempo entre la fecha de llegada a la emergencia y retorno a la estación (DT4) por tipo de emergencia y tipo de vehículo.

**CÓDIGO EN R**

```
#dt4
      set.seed(3246)
      pb=txtProgressBar(max = 5*13,style = 3)
      tipVehiNames=names(disp_vehi)[-1]
      for (te in 1:5) {
        for (tv in 1:13) {
          indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]]
      & partes_nuevo$TipEmergencia==te)

      partes_nuevo[indPartes,dt4:=rkernel(dt4_dens[[paste('t',te,sep
      ='_')]][[tipVehiNames[[tv]]]],N = length(indPartes),T)]
          setTxtProgressBar(pb,(te-1)*13+tv)
        }
      }
      close(pb)
      rm(pb,te,tv,indPartes)
```

## 4.7. Simulación de emergencias

La simulación que se genera pertenece a eventos discretos, donde cada evento se toma como un suceso que cambia las variables de estado del sistema.

**Tipos de eventos**

Los tipos de eventos con los que se cuenta son:

- Lanzamiento del vehículo hacia emergencia (tipo 1)
  Este evento ocurre cuando comienza el delta DT1, donde se reserva el vehículo para la atención a la emergencia.
- Reingreso del vehículo a la estación (tipo 2)
  Este evento ocurre cuando el vehículo llega a la estación y se libera para poder atender otra emergencia.

**CÓDIGO MATLAB**

```
Código para inicialización de eventos tipo 1

Eventos=[partes2(:,1),ones(length(partes2),1),(1:length(partes2)).
'];
```

**Inicialización de variables**

- Cargar información de las variables presimuladas
- Todos los vehículos comienzan con estado disponible y se almacenan en la matriz “**esActual**”
- Se almacena el índice del evento actual en “**evActual**” y se inicializa con el valor de 1.

## Simulación del sistema

Mientras que "**evActual**" sea menor que el número de eventos que existen y el tiempo del evento actual sea menor que el tiempo máximo de simulación, se siguen las siguientes condiciones:

Si el tipo de **evento actual** es **1**, se halla la compañía más cercana que tenga el tipo de vehículo que se requiere.

- Si la compañía existe, se siguen los siguientes pasos:

| Paso | Descripción |
|---|---|
| 1 | Se resta 1 vehículo de ese tipo a la compañía. |
| 2 | Se almacena la compañía dentro del registro correspondiente. |
| 3 | El tiempo de despacho será igual al tiempo del evento más el DT1 de ese registro. |
| 4 | El tiempo de salida es el tiempo de despacho más DT2. |
| 5 | Se calcula el DT3 según la compañía asignada y el bloque horario al que pertenece el tiempo de salida. |
| 6 | Tiempo de llegada es igual al tiempo de salida + DT3 calculado. |
| 7 | El tiempo de retirada es igual al tiempo de llegada más el DT4. |
| 8 | Se calcula el DT5 según la compañía asignada y el bloque horario al que pertenece el tiempo de retirada. |
| 9 | El tiempo de reingreso es igual al tiempo de retirada + DT5. |
| 10 | Se genera un nuevo evento tipo 2 con un tiempo igual al tiempo de reingreso, y se coloca en la lista de eventos en su posición correspondiente. |

- Si no existe la compañía: El evento se mueve al siguiente tiempo (dentro de la prioridad de eventos) y el tiempo de ese evento se cambia al que corresponde.

Si el tipo del **evento actual** es **2**:

- Se agrega un vehículo de ese tipo a la compañía de origen.
- Se deja de lado el evento cerrado y "**evActual**" toma el siguiente índice.
- Por último, se eliminan los registros que no se llegaron a atender.

A continuación, se presenta el código utilizado para la simulación.

**CÓDIGO EN MATLAB**

```
Código de simulación del sistema

esActual=EstadoInicial;
evActual=1;
partes2(:,[14,15])=0;
while evActual<size(Eventos,1) && Eventos(evActual,1)<=max
    if Eventos(evActual,2)==1
        parteIndex=Eventos(evActual,3);
```

```
        distTemp=distancias_ida(:,partes2(parteIndex,6));
        [~,I]=sort(distTemp);
        ciaTemp=I(esActual(I,partes2(parteIndex,9))>0);
        if ~isempty(ciaTemp)
            ciaTemp=ciaTemp(1);
esActual(ciaTemp,partes2(parteIndex,9))=esActual(ciaTemp,partes2(p
arteIndex,9))-1;
            partes2(parteIndex,15)=cias(ciaTemp,1);
partes2(parteIndex,10)=partes2(parteIndex,10)+Eventos(evActual,1);
partes2(parteIndex,11)=partes2(parteIndex,11)+partes2(parteIndex,1
0);
t1=genNumeros(velDist_ida{ceil(mod(partes2(parteIndex,11),7)*6)},1
)*distancias_ida(ciaTemp,partes2(parteIndex,6));
            partes2(parteIndex,12)=partes2(parteIndex,11)+t1;
partes2(parteIndex,13)=partes2(parteIndex,13)+partes2(parteIndex,1
2);
t2=genNumeros(velDist_vuelta{ceil(mod(partes2(parteIndex,13),7)*6)
},1)*distancias_vuelta(ciaTemp,partes2(parteIndex,6));
            partes2(parteIndex,14)=partes2(parteIndex,13)+t2;
            i1=evActual+1;
            while Eventos(i1,1)<partes2(parteIndex,14)
                i1=i1+1;
            end
            Eventos(i1+1:end+1,:)=Eventos(i1:end,:);
            Eventos(i1,:)=[partes2(parteIndex,14),2,parteIndex];
        else
            i1=evActual+1;
            while Eventos(i1,1)<=Eventos(evActual,1)
                i1=i1+1;
            end
            Eventos(evActual,1)=Eventos(i1,1);
            t1=Eventos(evActual,:);
            Eventos(evActual:i1-2,:)=Eventos(evActual+1:i1-1,:);
            Eventos(i1-1,:)=t1;
            evActual=evActual-1;
        end
        clear i1 t1 t2
    elseif Eventos(evActual,2)==2
        parteIndex=Eventos(evActual,3);
        ciaTemp=partes2(parteIndex,15);
        ciaTemp=find(cias(:,1)==ciaTemp,1);
        vehiTemp=partes2(parteIndex,9);
        esActual(ciaTemp,vehiTemp)=esActual(ciaTemp,vehiTemp)+1;
        clear vehiTemp ciaTemp
    elseif Eventos(evActual,2)==3
esActual(floor(Eventos(evActual,3)),round(mod(Eventos(evActual,3),
1)*100))=...
esActual(floor(Eventos(evActual,3)),round(mod(Eventos(evActual,3),
1)*100))-1;
    elseif Eventos(evActual,2)==4
esActual(floor(Eventos(evActual,3)),round(mod(Eventos(evActual,3),
1)*100))=...
esActual(floor(Eventos(evActual,3)),round(mod(Eventos(evActual,3),
1)*100))+1;
    end
    evActual=evActual+1;
end
```

## 4.8. Validación del modelo

A continuación, se explicará los pasos utilizados para realizar la validación del modelo simulado.

### 4.8.1. Separación de deltas de tiempo

Los deltas de tiempo generados de la simulación son calculados a partir de las fechas de despacho, salida, llegada a la emergencia, retirada y reingreso a la estación. Debido al origen de los datos que se utilizan para simular DT3 y DT5, a estos deltas de tiempo se les aplica una depuración utilizando la función "**tukeyOutRange**". Además, para realizar la validación de los tiempos simulados, estos se comparan con los tiempos reales de la data histórica, a los cuales también se les aplicó una depuración de borrado de outliers.

La similitud entre las diferencias de tiempo simulados y los reales se puede observar en la Figura 27, donde los datos simulados son representados por la línea de color **azul** y los reales por la de color **naranja**. En algunos gráficos se aprecia una diferencia en la amplitud de ciertos picos; sin embargo, esto se debe a la naturaleza de la distribución de Kernel y su base Normal que varía según la cantidad de datos.

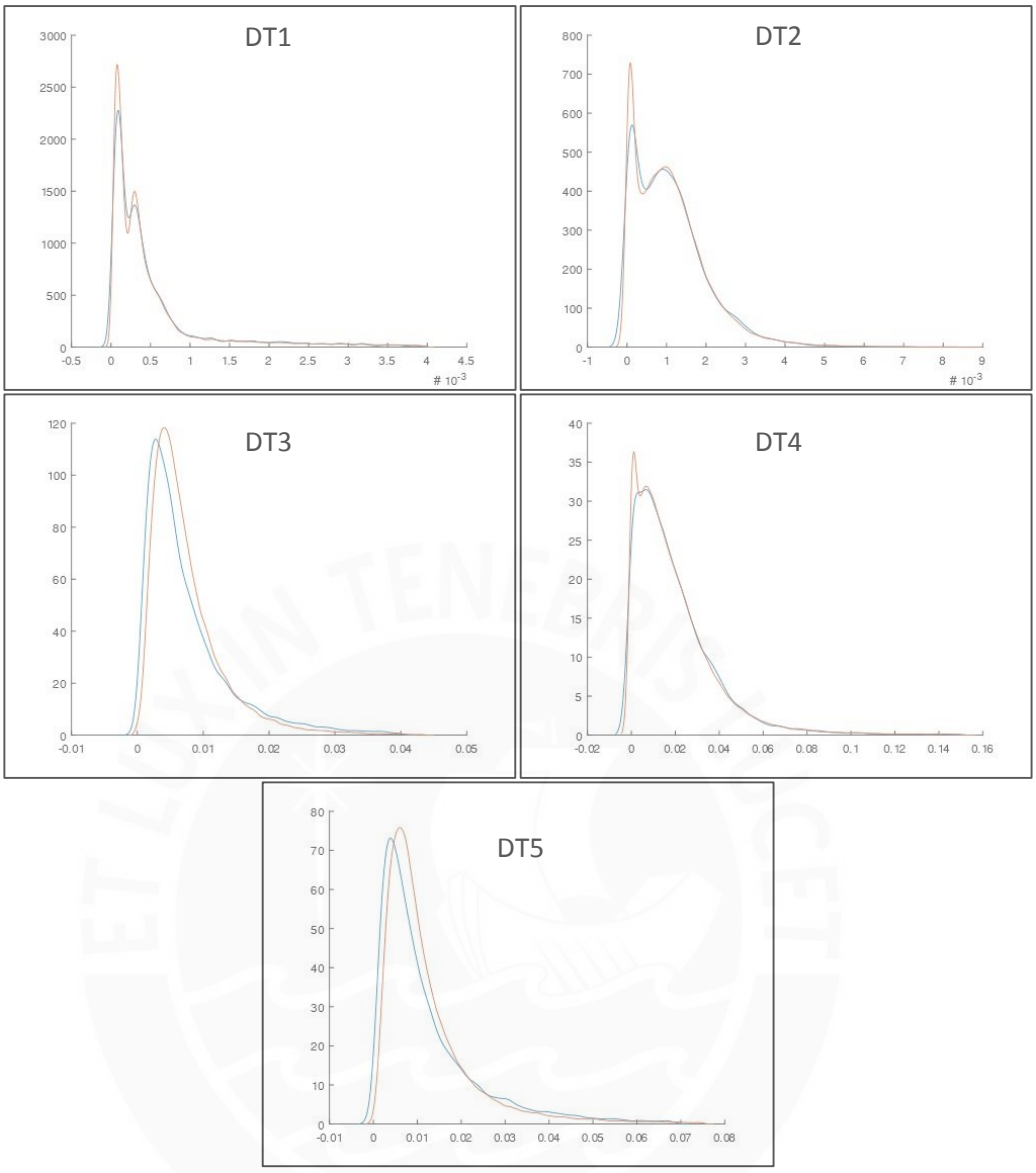



*Figura 27: DT's real vs simulado*

*Elaboración propia*

### 4.8.2. Test Kolmogorov-Smirnov

Para que se pueda validar que las diferencias de tiempo (DT1, DT2, DT3, DT4 y DT5) reales y los simulados tienen como origen a la misma distribución, se utilizará la prueba de Kolmogorov – Smirnov para dos muestras, donde las hipótesis serán las siguientes:

$$H_0 = La\ data\ de\ la\ muestra\ 1\ y\ 2\ provienen\ de\ la\ misma\ distribución\ continua.$$

$$H_1 = La\ data\ de\ la\ muestra\ 1\ y\ 2\ provienen\ de\ diferentes\ distribuciones\ continuas.$$

**CÓDIGO EN MATLAB**

```
disp('------------Test de Kolmogorov-Smirnov')
```

```
for dtn=1:5

disp(strcat('dt',num2str(dtn),':',num2str(kstest2(dt{dtn,1}(:,2),d
t{dtn,2}(:,2),'alpha',0.05)),': ', ...

num2str(floor(mean(dt{dtn,1}(:,2))*24*60)),'m',num2str(floor(mod(m
ean(dt{dtn,1}(:,2))*24*60,1)*60)),'s-', ...

num2str(floor(mean(dt{dtn,2}(:,2))*24*60)),'m',num2str(floor(mod(m
ean(dt{dtn,2}(:,2))*24*60,1)*60)),'s:', ...
        num2str(((mean(dt{dtn,1}(:,2))-
mean(dt{dtn,2}(:,2)))/mean(dt{dtn,2}(:,2)))*100),'%'))
end
clear x1 x2
```

Los resultados obtenidos al utilizar el código de validación de Kolmogorv - Smirnov son los siguientes:

```
----------Test de Kolmogorov-Smirnov
dt1:0:  0m44s       -       0m43s  :       0.97121%
dt2:0:  1m35s       -       1m35s  :       -0.42467%
dt3:1:  10m43s      -       10m44s:       -0.083434%
dt4:0:  28m4s       -       27m46s:       1.0722%
dt5:1:  17m24s      -       17m35s:       -1.0858%
```

Se observa que los deltas de tiempo donde no se rechaza la hipótesis nula con un 95% de significancia son el DT1, DT2 y DT4. Por otro lado, en los deltas DT3 y DT5 se rechaza la hipótesis nula con 5% se significancia. Por otro lado, para estos últimos DT's, no se puede afirmar que provienen de la misma distribución de los datos iniciales, sin embargo, el promedio de cada uno coincide con el original. Además, estos resultados pueden haberse visto afectados por el tamaño de la muestra.

### 4.8.3. Separación en lotes

Al realizar una simulación, usualmente en la línea de tiempo se observan dos intervalos diferenciados, los cuales son el periodo de calentamiento y el estado estable (1.7.4). Para este caso, utilizando el método visual, se considera que el periodo de calentamiento es de 20 días, por lo que se decide truncar los registros en ese punto. En la Figura 28 se puede apreciar el promedio de los DT's a través del tiempo (días).

Como siguiente paso, se realiza la separación en batches (o lotes) del tiempo total de simulación luego del truncado, para que, posteriormente, se pueda utilizar la prueba de intervalos de confianza. En primer lugar, se separa el tiempo total en 186 lotes para cada delta de tiempo, para obtener por lo menos 50 observaciones por lote, y se halla el promedio de cada lote.

**CÓDIGO EN MATLAB**

Código para truncar registros

```
truncar=20;
for dtn=1:5
    dt{dtn,1}=dt{dtn,1}(dt{dtn,1}(:,1)>=truncar,:);
end
```

Código para la separación en batches

```
nBatch=floor(length(dt{1,1})/50);
batches=zeros(nBatch,5);
batches2=cell(nBatch,5);
batchSize=(tSimu-truncar)/nBatch;
for dtn=1:5
    pTemp=dt{dtn,1};
    for batch=1:nBatch

batches2{batch,dtn}=pTemp(pTemp(:,1)<=truncar+batchSize*batch &
pTemp(:,1)>truncar+batchSize*(batch-1),2)*24*60*60;
        batches(batch,dtn)=mean(batches2{batch,dtn});
    end
end
 clear batchSize dtn pTemp batch
```

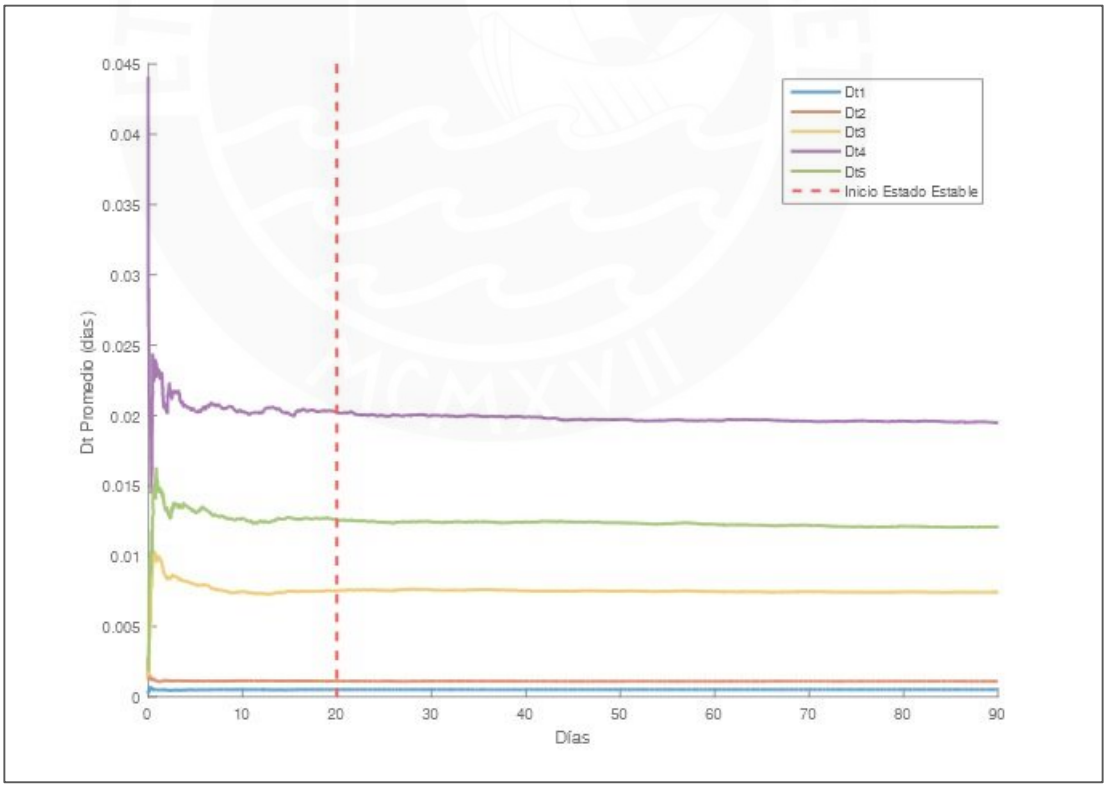



*Figura 28: Promedio de DT's*

*Elaboración propia*

Posteriormente, se analiza la correlación que existe entre los valores promedio de cada lote. En específico, se revisa que la correlación del lag-1 sea menor o igual que 0.2 (Banks, 2010).

**CÓDIGO EN MATLAB**

Código de correlación

```
lag=zeros(5,1);
ro=zeros(5,1);
for dtn=1:5
    a=autocorr(batches(:,dtn),nBatch-1);
    lag(dtn)=find(a<=0.2,1,'first')-1;
    ro(dtn)=a(2);
end
clear a
```

En la Figura 29 se observa el correlograma con lag -0 a -40, y se comprueba que la correlación es menor a 0.2.

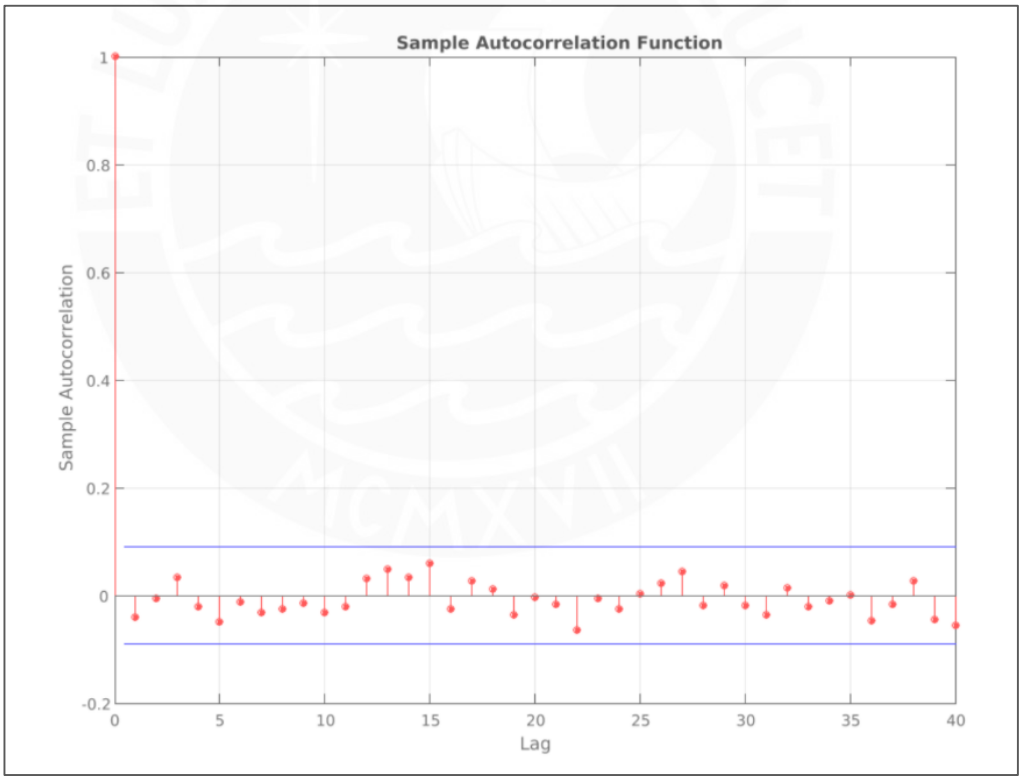



*Figura 29: Correlograma de la simulación*

*Elaboración propia*

Dado que la correlación del lag-1 cumple con el requisito mencionado anteriormente, se procede a reagrupar los lotes en k=30 nuevos lotes. Además, se realiza una validación adicional para corroborar la independencia de los lotes.

**CÓDIGO EN MATLAB**

```
Código para reagrupación de lotes

k=30;
batches3=zeros(k,5);
bsize=floor(length(batches)/k);
for dtn=1:5
    for batch=1:k
        batches3(batch,dtn)=mean(batches(bsize*(batch-
1)+1:bsize*batch,dtn));
    end
end
clear dtn batch

Código de test de Independencia

C=sqrt((k*k-1)/(k-2))*(ro.'+((batches3(1,:)-
mean(batches3)).^2+(batches3(end,:)-
mean(batches3)).^2)./(2*sum((batches3-
repmat(mean(batches3),k,1)).^2)));
disp(C< norminv(0.975,0,1))
```

Los resultados muestran que los lotes no presentan correlación significativa, por lo que se procede a realizar al análisis de intervalos de confianza para las medias.

*Tabla 21: Valor del estadístico por DT*

| Dt | Estadístico ($<Z_{0.975}$) |
|---|---|
| 1 | 0.168 |
| 2 | -0.0134 |
| 3 | 1.2047 |
| 4 | 0.4247 |
| 5 | 0.7376 |

*Elaboración propia*

### 4.8.4. Test de Intervalos de confianza para las medias

Para realizar esta prueba, se hará uso del promedio y las desviaciones estándar por delta de tiempo de la última separación de lotes. Las hipótesis son las siguientes:

$$H_0 = La\ medias\ de\ la\ muestra\ 1\ y\ 2\ son\ iguales.$$

$$H_1 = La\ medias\ de\ la\ muestra\ 1\ y\ 2\ no\ son\ iguales.$$

**CÓDIGO EN R**

```
Código para el intervalo de confianza de las medias

disp('------------Test de Intervalos de Confianza')
for dtn=1:5
    rango=tinv(0.975,length(batches3(:,dtn))-
```

```
1)*std(batches3(:,dtn))/sqrt(length(batches3(:,dtn)));
disp(strcat('dt',num2str(dtn),':',num2str(mean(batches3(:,dtn))-
rango),'-
',num2str(mean(batches3(:,dtn))+rango),':',num2str(mean(dt{dtn,2}(
:,2))*24*60*60),':',num2str(abs(mean(dt{dtn,2}(:,2))*24*60*60-
mean(batches3(:,dtn)))<rango)))
end
clear rango
```

Los resultados se muestran a continuación, donde se observa el intervalo de confianza de cada delta de tiempo, el tiempo promedio de la data real y finalmente si este tiempo se encuentra dentro del intervalo de confianza, entonces no se rechaza la hipótesis nula.

```
----------Test de Intervalos de Confianza
dt1: 42.8183    -    45.3386    :43.6358    :1
dt2: 93.841     -    96.6795    :95.55      :1
dt3: 621.4756   -    656.049    :644.5027   :1
dt4: 1651.5718  -    1716.2332  :1666.577   :1
dt5: 1009.6661  -    1067.824   :1055.6285  :1
```

Lo expuesto anteriormente también se puede observar de manera gráfica en la Figura 30, donde los intervalos de confianza se representan mediante una línea azul y los valores reales mediante un punto rojo. Se refleja que en los cinco intervalos de tiempo el punto rojo se encuentra dentro del rango, lo que indica que no se cuenta con evidencia suficiente para afirmar que no son iguales.

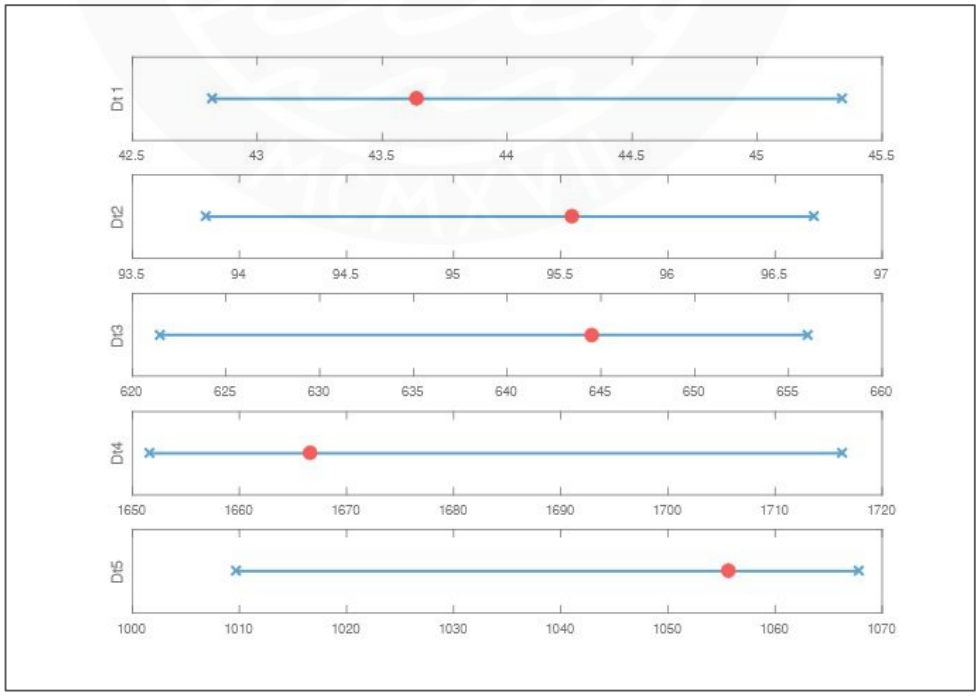



*Figura 30: Intervalos de confianza para DT's*
*Elaboración propia*

# CAPÍTULO 5. APLICACIÓN DEL MODELO Y PROPUESTAS DE MEJORA

En este capítulo se presentan las variables que se pueden modificar para realizar nuevas simulaciones y probar el impacto que generan en el sistema. Además, se muestran los resultados de las nuevas simulaciones y las propuestas de mejora encontradas en el sistema del Cuerpo General de Bomberos Voluntarios del Perú.

## 5.1. Preparación de variables

Dado que se aplicará el principio de ceteris paribus, todas las variables de entrada se cargan a partir de la data histórica y se modifica una a la vez. A continuación, se exponen las variables que pueden ser sujetas a modificaciones.

Variables modificables

- **Disponibilidad de vehículos por compañía**
  Se pueden incrementar o disminuir la cantidad de vehículos disponibles inicialmente en cada compañía, según su tipo.
- **Localización de compañías**
  Se puede proponer nuevas localizaciones de compañías, así como el cierre o traslado de otras.
- **Tiempo de respuesta dt1**
  Se puede medir el impacto del sistema de atención de llamadas al modificar el tiempo de respuesta ya sea por obtener una mejora en su proceso o, en caso contrario, la disminución del personal de atención.
- **Velocidad promedio de ida y vuelta**
  Se puede medir el impacto en el sistema al variar la velocidad promedio para el traslado de vehículos, lo cual puede ser debido a nuevas políticas viales o de tránsito y mejora en la cultura vial.
- **Calles cerradas por obras o eventos**
  Se puede modificar el mapa subyacente para representar el cierre de calles, y así poder medir el impacto que causa en el sistema.
- **Nuevos proyectos de infraestructura vial**
  De la misma manera que el punto anterior, es factible la modificación y creación de nuevas rutas y, de esta manera, representar la habilitación de nuevas calles.
- **Tiempo promedio entre emergencias**

Se puede modificar el tiempo promedio entre emergencias para medir el impacto en el sistema, en caso se evidencia un incremento en la incidencia de emergencias.

**Variable elegida para la nueva simulación**

Se ha elegido modificar la variable de localización de compañías, donde se busca hallar una nueva posición que disminuya el tiempo de atención promedio de la emergencia. Para esta variación se proponen tres métodos para hallar nuevas localizaciones explicados a continuación:

- **Método del centro de gravedad**

Este método brinda la posición para la nueva estación según el centro de gravedad de las emergencias. En ese sentido, se halla el promedio de las posiciones de todas las emergencias únicas. Se hace uso del siguiente código:

**CÓDIGO EN MATLAB**

Código para hallar el centro de gravedad

```
[~,I]=unique(Partes(:,1));
cias2=[cias2;mean(Partes(I,5:6))];
```

- **Método del centro de gravedad ponderado**

Al igual que en el punto anterior, se hallará la nueva posición según el centro de gravedad de las emergencias; sin embargo, en este caso, se ponderan las posiciones de las emergencias únicas según el tiempo de atención del primer vehículo. Se hace uso del siguiente código:

**CÓDIGO EN MATLAB**

Código para hallar el centro de gravedad ponderado

```
[~,I]=sort(Partes(:,10));
pTemp=Partes(I,:);
[~,I]=unique(pTemp(:,1));
pTemp=pTemp(I,:);
cias2=[cias2;(pTemp(:,10)-
pTemp(:,2)).'*pTemp(:,5:6)/sum((pTemp(:,10)-pTemp(:,2)))];
```

- **Método visual**

Este método consta de graficar el mapa de Lima y las estaciones que existen actualmente, y de manera visual se halla el posible lugar donde se pueda ubicar una nueva estación; por ejemplo, una posición donde no se tengan estaciones cerca. Se hace uso del siguiente código:

**CÓDIGO EN MATLAB**

```
Código para utilizar el método visual

x1=repmat(nodos(:,2),1,size(cias,1)).';
y1=repmat(nodos(:,3),1,size(cias,1)).';
x2=repmat(cias(:,2),1,size(nodos,1));
y2=repmat(cias(:,3),1,size(nodos,1));
[Dmin,Imin]=min(sqrt((x2-x1).^2+(y2-y1).^2));
figure
axis image
hold on
for i1=1:size(cias,1)
    scatter(nodos(Imin==i1,3),nodos(Imin==i1,2),'.')
end
hold off
```

En la Figura 31 se observa el resultado de la ejecución del código. Cada estación está representada por un asterisco blanco.

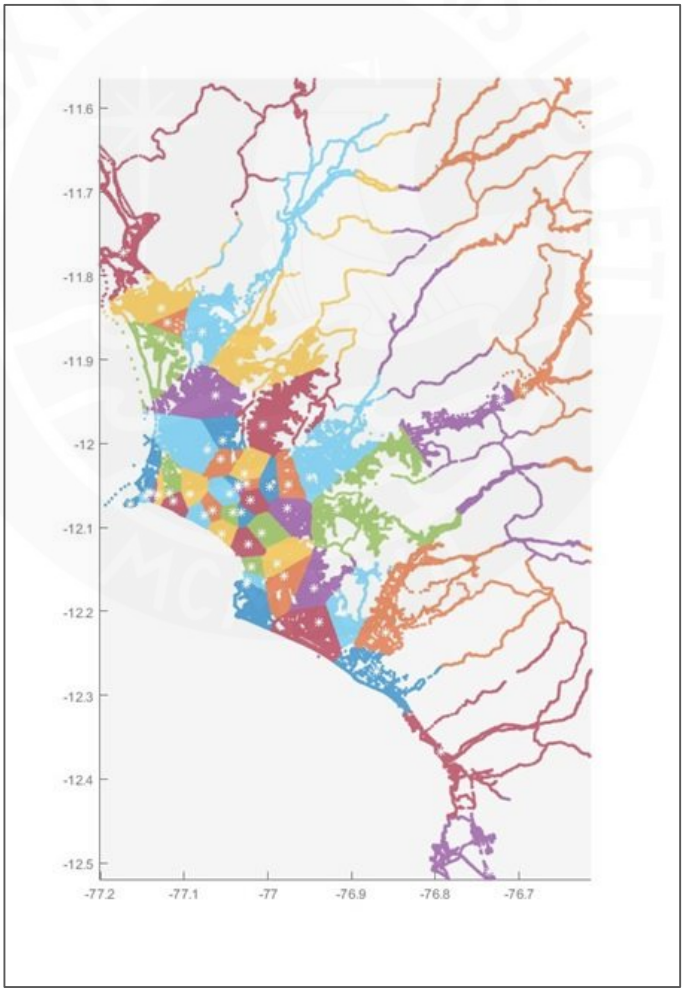



*Figura 31: Estaciones y zonificación según proximidad en Lima*

*Elaboración propia*

Valores seleccionados para la variable

Luego de ejecutar los códigos de cada método, se obtienen las siguientes posiciones:

*Tabla 22: Posiciones propuestas para nuevas estaciones*

| Método | Latitud | Longitud |
|---|---|---|
| Centro de gravedad | -12.0715 | -77.0275 |
| Centro de gravedad ponderado | -12.0688 | -77.0227 |
| Visual: Opción 1 | -12.2056 | -76.9908 |
| Visual: Opción 2 | -11.9996 | -77.1168 |

*Elaboración propia*

En la Figura 32 se muestran las posiciones propuestas por cada método, en el mapa de Lima.

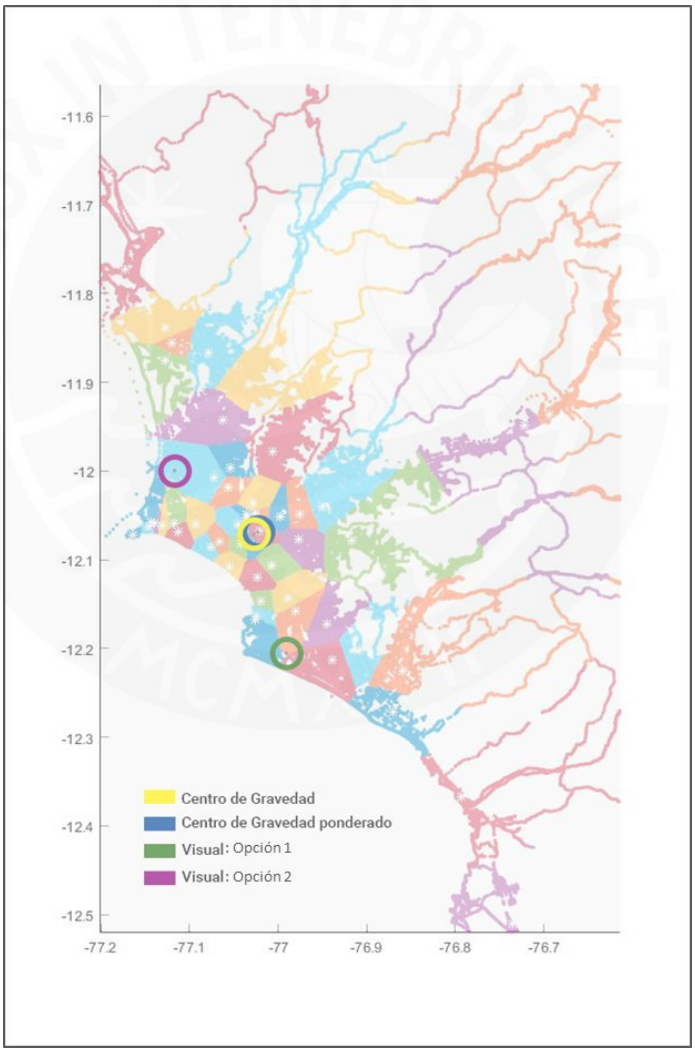



*Figura 32: Selección de nuevas estaciones*

*Elaboración propia*

## 5.2. Simulación del sistema con modificaciones

Dadas las modificaciones que se especificaron en la preparación de variables, se procede a simular el sistema con una variable modificada a la vez.

Para cada caso se agrega una compañía a la lista de compañías existentes. Luego, se calculan las rutas y distancias de ida y vuelta para la nueva compañía, y se adjuntan a las matrices correspondientes. Además, se ingresa la nueva compañía en el estado inicial del sistema. Para el caso de las compañías propuestas, cada una inicia con una ambulancia y dos autobombas, ya que, en promedio, las compañías cuentan con esos vehículos. Finalmente, se procede a realizar la presimulación y la simulación para cada uno.

## 5.3. Comparación y elección

Para la comparación y elección de la propuesta conveniente se utilizará el método de intervalos de confianza para el promedio de los tiempos de llegada a la emergencia.

*Tabla 23: Intervalos de confianza por propuesta*

| N° | Propuesta | Dt3 (s) | | | Dt5 (s) | | |
|---|---|---|---|---|---|---|---|
| | | Límite inferior | Límite superior | Actual | Límite inferior | Límite superior | Actual |
| 1 | Centro de gravedad | 621.49 | 644.94 | 644.50 | 990.42 | 1031.91 | 1055.63 |
| 2 | Centro de gravedad ponderado | 628.99 | 653.13 | | 988.17 | 1037.35 | |
| 3 | Visual: Opción 1 | 616.18 | 643.60 | | 991.56 | 1039.40 | |
| 4 | Visual: Opción 2 | 609.01 | 634.35 | | 966.61 | 1009.43 | |

*Elaboración propia*

En la tabla 23 se cuenta con los intervalos de confianza del DT3 y DT5 de cada una de las propuestas. Se evaluará elegir las propuestas cuyo intervalo de confianza sea menor que los demás.

En la Figura 33 y 34 se muestra la comparación entre las propuestas y el valor de la media real para DT3 y DT5 respectivamente. Los intervalos de confianza de las 4 propuestas se representan en líneas horizontales, cuyos inicios y fines se denotan por un aspa, y el tiempo promedio actual real se representa por la línea vertical.

Como se observa en la Figura 33, las propuestas cuyo intervalo de confianza del DT3 es significativamente menor que el tiempo real, son las sugeridas por las opciones del método visual (propuesta 3 y 4), ya que los intervalos de confianza se encuentran antes que el tiempo de la media actual. Por otro lado, como se observa en la Figura 34, todas las propuestas son significativamente menores que el tiempo DT5 real.

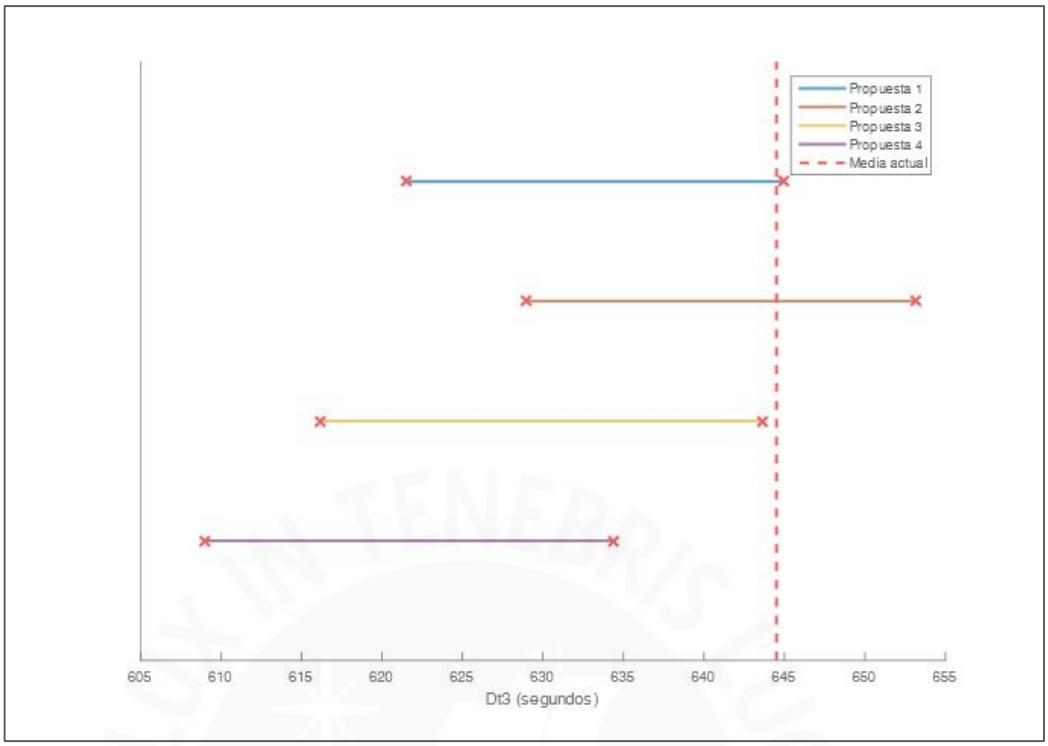



*Figura 33: Intervalos de confianza para resultados de DT3*

*Elaboración propia*

Las posibles propuestas a elegir son la 3 y 4; sin embargo, la elegida será la 4 dado que es significativamente menor que el tiempo DT3 real y la propuesta 3 presenta un escenario no tan favorable para DT5.

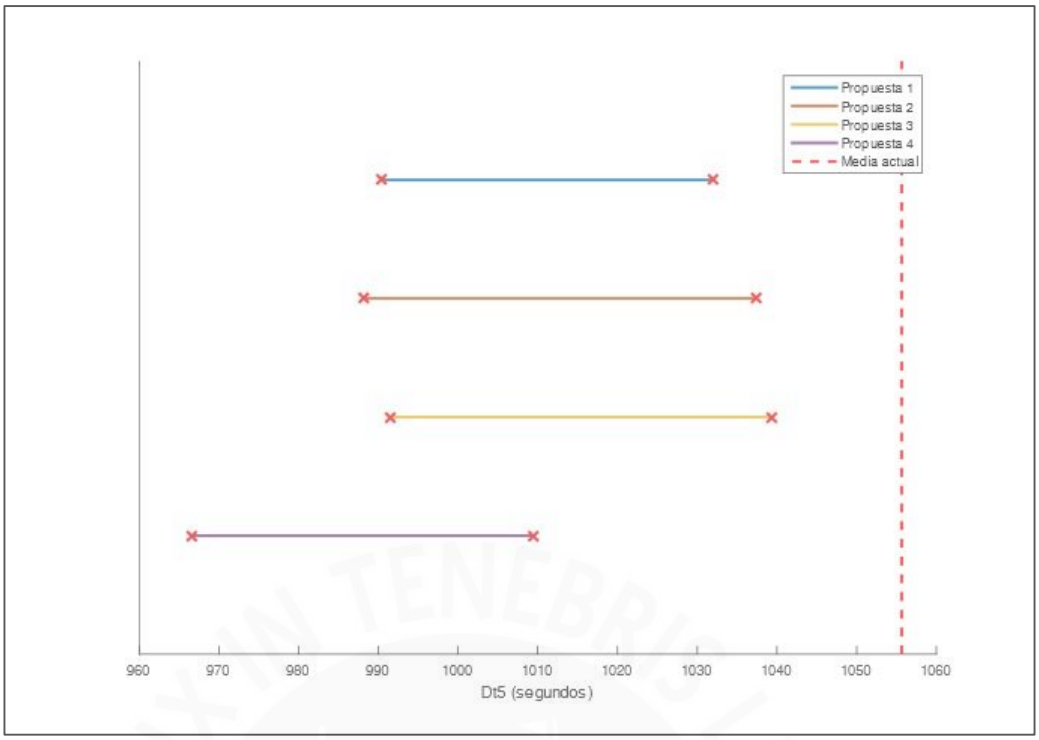



*Figura 34: Intervalos de confianza para resultados de DT5*

*Elaboración propia*

# CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

Durante el desarrollo y aplicación del caso de estudio se han observado oportunidades de mejora que podrían ayudar a facilitar el proceso de simulación y obtener resultados más precisos. A continuación, se recopilarán las conclusiones y recomendaciones más relevantes.

## 6.1. Conclusiones

- Se logró encontrar una propuesta de mejora significativa gracias al modelo de simulación planteado, reduciendo el tiempo de atención de 13.06 minutos a 12.68 minutos (llegada al lugar de la emergencia).
- El modelo de simulación de eventos discretos es una herramienta que permite hacer pruebas y evaluar escenarios de manera bastante flexible, eficiente, de bajo costo y libre de riesgo.
- Gracias a la flexibilidad de los programas utilizados (R y Matlab), se pudo manejar gran cantidad de datos de manera matricial sin causar lentitud en el sistema. Además, permitió ejecutar códigos paralelamente, y así obtener resultados en menor tiempo.
- Durante el desarrollo de la tesis, se implementaron distintos algoritmos para la estimación del tráfico hasta que se eligió el método mostrado dentro del capítulo 4. Sin embargo, cabe mencionar que, si se cuenta con la información necesaria, es posible implementar un programa lineal que disminuya el error del tráfico o, en su defecto, una heurística que permita hallar el tráfico por zonas.
- Uno de los supuestos asumidos es la independencia entre las variables a simular dentro del sistema, lo cual no es necesariamente cierto, por lo que se concluye que, para un mejor análisis, se puede tomar la interacción entre las variables.
- Se pueden analizar escenarios diferentes a los propuestos en el capítulo 5, por ejemplo, la redistribución de recursos entre compañías (vehículos) y la capacitación de los bomberos para reducir el tiempo de preparación (DT2). Ambas variables pueden tener un fuerte impacto en el tiempo total de respuesta.

## 6.2. Recomendaciones

- Se sugiere realizar un análisis de utilización de recursos, dado que este es otro frente importante para optimizar.
- Se sugiere mejorar la toma de datos de los registros, ya que se han encontrado partes con información incoherente, como son emergencias sin latitud y longitud, falta de precisión y veracidad en las fechas de registro, envío, salida, llegada y retorno de la emergencia.
- Para que la simulación y los ruteos se ajusten a la realidad con mayor precisión, se recomienda utilizar data real del tráfico que sea actualizada paulatinamente.
- Se debe mejorar el control de las fallas de vehículos, ya que esto ayudaría a tener contabilizado con un número cercano a la realidad los recursos con los que cuenta cada estación de bomberos. Además, permite realizar un mejor análisis de capital estancado (activos sin uso) e implementar fallas y mantenimiento dentro del modelo.
- Es recomendable contar con data histórica del cierre de calles dentro de Lima.
- Para la futura recolección de datos, es recomendable registrar la ruta exacta tomada por cada vehículo, esto para poder lograr una mejor estimación del tráfico y poder tener redundancias que permitan la corrección de errores en los registros.
- Se recomienda reestructurar los protocolos utilizados por el Cuerpo General de Bomberos Voluntarios del Perú, dado que en la actualidad estos no son respetados.

## Referencias Bibliográficas


(MINSA), M. d. (2008). *www.minsa.gob.pe*. Obtenido de https://www.minsa.gob.pe/portada/Especiales/2009/accidentes_transito/situacion.html

Alexopolus, C., & Seila, A. (1996). *Impementing the Batch Means Method in Simulation Experiments.* California: Proceedings of the 28th conference on Winter simulation, 214 - 221.

Association, N. F. (2014). *NFPA 1720, standard for the organization and deployment of fire suppression operations, emergency.* Quincy.

Banks, J. (2010). *Discrete-event system simulation.* Montreal: Prentice Hall.

Chakravarti, L. (1967). *Handbook of Methods of Applied Statistics.* New York: John Wiley.

Cormen, T. H. (2009). *Introduction to algorithms.* Boston: McGraw-Hill.

Corporation Microsoft, & Stephen Weston. (2017). *DoSNOW: Foreach Parallel Adaptor for the 'Snow' Package.* Obtenido de https://CRAN.R-project.org/package=doSNOW.

Cowley, R. (1975). A total emergency medical system for the state of Maryland. *Md State Med J*, 37-45.

Cuerpo General de Bomberos Voluntarios del Perú (CGBVP). (2016). Lima, Perú. Obtenido de http://www.bomberosperu.gob.pe

Departamento de Matemática de la Universidad Estatal de Kansas. (8 de agosto de 2018). *Universidad Estatal de Kansas*. Obtenido de https://www.math.ksu.edu/~dbski/writings/haversine.pdf

Dijkstra, E. (1959). A Note on Two Problems in Connexion with Grpahs. *Numerische Mathematik 1*, 269 - 271.

Dowle, M. & Arun S. (2018). *Data.table: Extension of 'Data.frame'.* Obtenido de https://CRAN.R-project.org/package=data.table.

Gómez, D. L. (2015). Análisis espacial de los accidentes de tráfico con víctimas mortales en carretera en España, 2008-2011. *Gaceta Sanitaria*, 24-29. doi:https://doi.org/10.1016/j.gaceta.2015.02.009

Government, D. f. (2015). *Fire Incidents Response Times: April 2014 to March 2015, England.* Inglaterra.

Gramacki, A. (2018). *Nonparametric Kernel Density Estimation and Its Computational Aspects.* Polonia: Springer.

Grolemund, G. ,. (2011). *“Dates and Times Made Easy with lubridate.” Journal of Statistical Software 40 (3): 1–25.* Obtenido de http://www.jstatsoft.org/v40/i03/

Gujarati, D., & Porter, D. (2010). *Econometría.* México D. F.: McGraw-Hill.

INESEM, R. d. (Noviembre de 2016). Obtenido de https://revistadigital.inesem.es/biosanitario/hora-de-oro/

Jungnickel, D. (2008). *Graphs, networks and algorithms.* Berlín: Springer.

Law, A., & Kelton, W. (1991). *Simulation modeling and analysis.* New York: McGraw Hill.

Lyman, M., Willis, F., & James, R. (1940). *Plane and Spherical Trigonometry.* New York: McGraw-Hill.

Martinez, F., & Quetglás, G. (2003). *Introducción a la programación estructurada en C.* Valencia: Universitat de Valencia.

Perú, J. d. (2016).

Schmeiser, B. (1982). *Batch Size Effects in the Analysis of Simulation Output.* Indiana: Operation Research Society of America, 556 - 558.

Sedgewick, R. (1998). *Algortimos en C++.* Delaware: Addison-Wesley.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis.* London: Chapman and Hall.

Snedecor, G., & Cochran, W. (1989). *Statistical Methods.* Iowa: Iowa State University Press.

The journal of the Medical Society of New Jersey. (1987). *New Jersey medicine, Vol 74*(11).

Tukey, J. (1977). *Exploratory Data Analysis.* Washington: Addison-Wesley.

Venables, W. &. (2002). *Modern Applied Statistics with S. Fourth.* New York: Springer.

Vrac, M. &.-A. (2012). *CDFt: Statistical Downscaling Through Cdf-Transform*. Obtenido de https://CRAN.R-project.org/package=CDFt

Wackerly, D., Mendenhall, W., Scheaffer, R., & García, H. (2009). *Estadística matemática con aplicaciones.* México D.F.: Cengage Learning Editores.

Wei Lam, S., Zhang, Z., Oh, H., Ng, Y., Wah, W., & Hock Ong, M. (2014). Reducing ambulance response times using discrete. *Prehosp Emerg Care*. doi:http://dx.doi.org/10.3109/10903127.2013.836266

Wickham, H. (2016). *Ggplot2: Elegant Graphics for Data Analysis.* New York: Springer-Verlag.

Zhang, Y. &. (2013). Police patrol districting method and simulation evaluation using agent-based model & GIS. *Secur Inform*, 2-7. doi:https://doi.org/10.1186/2190-8532-2-7

**Pontificia Universidad Católica del Perú**

**Facultad de Ciencias e Ingeniería**

# ELABORACIÓN DE UN SIMULADOR PARA EL SISTEMA DE ATENCIÓN A EMERGENCIAS BRINDADA POR EL CUERPO GENERAL DE BOMBEROS VOLUNTARIOS DEL PERÚ APLICADO EN LIMA Y CALLAO

## ANEXOS

Presentado por:

**Hector Mattos Galarza**

**Mónica Milagros Huayta Durand**

**ASESOR: Jonatán Edward Rojas Polo**

Lima, noviembre de 2018

## Anexos

### Anexo 1. Código para hallar los diferenciales de tiempo

**CÓDIGO EN R**

**Código de deltas de tiempo**

```
dt2=[Partes(:,8),Partes(:,9)-Partes(:,8),Partes(:,17)];
dt3=[Partes(:,[9,3,7,5,6,16]),Partes(:,10)-
Partes(:,9),PartesAprx(:,1)];
dt4=[Partes(:,10),Partes(:,11)-
Partes(:,10),floor(Partes(:,4)/1000000),Partes(:,17)];
dt5=[Partes(:,[11,3,7,5,6,16]),Partes(:,12)-
Partes(:,11),PartesAprx(:,1)];
```

**Anexo 2. Código para obtener el grafo de Lima**

**CÓDIGO EN R**

```
library(data.table)
library(ggplot2)
setwd('D:/Archivos/Tesis/Data')
nodos=fread('nodosPeru.dat',colClasses = 'numeric')
arcos=fread('arcosPeru.dat')
#Quedarnos con Lima----
nodos=nodos[between(lat,-12.5196,-11.5653) & between(lon,-77.2010,-76.6144),]
arcos=arcos[nodo1%in%nodos$ID & nodo2%in%nodos$ID,]
arcos=arcos[hw%in%c('',
                    'residential',
                    'secondary',
                    'tertiary',
                    'service',
                    'primary',
                    'unclassified',
                    'living_street',
                    'trunk',
                    'motorway_link',
                    'secondary_link',
                    'motorway',
                    'primary_link',
                    'trunk_link',
                    'corridor',
                    'tertiary_link',
                    'road'),]
#Cambiar IDs----
nodos[,ID_new:=1:.N]
arcos[,nodo1:=nodos[match(nodo1,ID),ID_new]]
arcos[,nodo2:=nodos[match(nodo2,ID),ID_new]]
#Busqueda en anchura----
#Eliminar nodos que no tienen arcos
nodos=nodos[ID_new%in%arcos$nodo1 | ID_new%in%arcos$nodo2,]
nodos[,bfs1:=NA_real_]
nodos[,iter:=NA_real_]
analizado=numeric(0)
iteracion=numeric(0)
g_actual=0
arcos_temp=arcos
pb=txtProgressBar(min = 0,max = nrow(nodos),style = 3)
while (max(table(nodos$bfs1))<=sum(is.na(nodos$bfs1))) {
  g_actual=g_actual+1
  iterar=T
  frontera=nodos[which(is.na(bfs1))[1],ID_new]
  iteracion=c(iteracion,1)
  while(length(frontera)>0){
```

```
    analizado=c(analizado,frontera)
    frontera_temp=c(arcos_temp$nodo2[arcos_temp$nodo1%in%frontera],arcos_temp
$nodo1[arcos_temp$nodo2%in%frontera])
    arcos_temp=arcos_temp[!arcos_temp$nodo1%in%frontera & !arcos_temp$nodo2%i
n%frontera]
    frontera=frontera_temp
    frontera=intersect(frontera,setdiff(frontera,analizado))
    iteracion=c(iteracion,rep(last(iteracion)+1,length(frontera)))
    setTxtProgressBar(pb,length(analizado))
  }
  analizado=c(analizado,frontera)
  nodos[ID_new%in%analizado & is.na(bfs1),bfs1:=g_actual]
  nodos[ID_new%in%analizado & is.na(iter),iter:=iteracion[match(ID_new,analiz
ado)]]
  print(g_actual)
  print(sum(nodos$bfs1==g_actual,na.rm = T))
}
#graficar
ggplot(nodos[match(analizado,ID_new)][iter<1000],aes(x=lon,y=lat,col=iter))+g
eom_point(size=1,shape=20)+scale_color_gradientn(colours = rainbow(5))
#Eliminar nodos que no forman parte del grafo principal(#1)----
nodos=nodos[bfs1==1,]
nodos[,ID:=ID_new]
nodos[,ID_new:=NULL]
arcos=arcos[nodo1%in%nodos$ID & nodo2%in%nodos$ID,]
#Guardar----
fwrite(arcos,'arcosLima.dat',col.names = T)
fwrite(nodos,'nodosLima.dat',col.names = T)
```

## Anexo 3. Cantidad de vehículos disponibles en el estado inicial

*Tabla: Cantidad de vehículos iniciales*

| | TIPO DE VEHÍCULO | | | | | | | | | | | | | TOTAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N° de CIA | AMB | AUX | CIS | ELE | ESC | M | MAT | MED | QUIMI | RES | SAMU | SNOR | USA | |
| 1 | 1 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 4 |
| 3 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 1 | 1 | 0 | 7 |
| 5 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| 7 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| 9 | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 6 |
| 10 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 6 |
| 11 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 |
| 13 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 14 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 5 |
| 15 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 3 |
| 16 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 4 |
| 18 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 21 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 5 |
| 28 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 7 |
| 32 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 5 |
| 34 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 36 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 4 |
| 60 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 65 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 75 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 83 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 96 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 4 |
| 100 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 5 |
| 105 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 106 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 5 |
| 115 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 |
| 120 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| 121 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| 124 | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 6 |
| 125 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 127 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| 129 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 133 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 134 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 4 |
| 138 | 1 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 7 |
| 150 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 155 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 160 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 161 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 163 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 164 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 168 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 169 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 176 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 184 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 202 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 |
| 207 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

*Elaboración propia*

**Anexo 4. Código de matrices de distancia ida y retorno**

**CÓDIGO EN R**

```
#Matriz de distancia ida----
      dist_ida=matrix(nrow = nrow(nodos),ncol = nrow(cias))
      for (i in 1:51) {
        print(i)
        ady=fread(paste0('Adyacencias/Ida_',cias$ID_nodo[i],'.dat'))
        dist=data.table(index=1:nrow(nodos))
        dist[,y:=ady[match(nodos$ID,nodo),dist]]

      dist[,x:=coord2m(nodos$lat,nodos$lon,cias[ID==cias$ID[i],lat],cias[ID==
      cias$ID[i],lon])]
        fit=rlm(y~x,data=dist[!is.na(y) & !is.infinite(y),])
        dist[is.na(y) | is.infinite(y),y:=predict(fit,dist[is.na(y) |
      is.infinite(y),])]#Imputación por lm robusta
        dist_ida[,i]=dist$y
      }
      rm(dist,i,ady,fit)


#Matriz de distancia ret----
      dist_ret=matrix(nrow = nrow(nodos),ncol = nrow(cias))
      for (i in 1:51) {
        print(i)
        ady=fread(paste0('Adyacencias/Ret_',cias$ID_nodo[i],'.dat'))
        dist=data.table(index=1:nrow(nodos))
        dist[,y:=ady[match(nodos$ID,nodo),dist]]

      dist[,x:=coord2m(nodos$lat,nodos$lon,cias[ID==cias$ID[i],lat],cias[ID==
      cias$ID[i],lon])]
        fit=rlm(y~x,data=dist[!is.na(y) & !is.infinite(y),])
        dist[is.na(y) | is.infinite(y),y:=predict(fit,dist[is.na(y) |
      is.infinite(y),])]#Imputación por lm robusta
        dist_ret[,i]=dist$y
      }
      rm(dist,i,ady,fit)
```

**Anexo 5. Código de la reconstrucción de la ruta más corta**

**CÓDIGO EN MATLAB**

Código para hallar las matrices de distancias

```
%% Hallar matriz de distancias y rutas desde cada cia
load('F:\Drive\Tesis Investiga\Carpeta Final\BD\Mapa.mat')
load('F:\Drive\Tesis Investiga\Carpeta Final\BD\cias.mat')
poolobj = parpool;
distancias_ida=zeros(length(cias),length(nodos));
rutas_ida=zeros(length(cias),length(nodos));
parfor i=1:length(cias)
    [dist,ruta]=mi_dijkstra(nodos,arcos_dir,cias(i,4));
    distancias_ida(i,:)=dist.';
    rutas_ida(i,:)=ruta.';
end
distAnt_ida=rutas_ida;
for i=1:length(cias)

distAnt_ida(i,~isnan(rutas_ida(i,:)))=coord2m(nodos(~isnan(rutas_i
da(i,:)),2),nodos(rutas_ida(i,~isnan(rutas_ida(i,:))),2),nodos(~is
nan(rutas_ida(i,:)),3),nodos(rutas_ida(i,~isnan(rutas_ida(i,:))),3
)).';
end
clear i
arcos_dir=fliplr(arcos_dir);
distancias_vuelta=zeros(length(cias),length(nodos));
rutas_vuelta=zeros(length(cias),length(nodos));
parfor i=1:length(cias)
    [dist,ruta]=mi_dijkstra(nodos,arcos_dir,cias(i,4));
    distancias_vuelta(i,:)=dist.';
    rutas_vuelta(i,:)=ruta.';
end
distAnt_vuelta=rutas_vuelta;
for i=1:length(cias)

distAnt_vuelta(i,~isnan(rutas_vuelta(i,:)))=coord2m(nodos(~isnan(r
utas_vuelta(i,:)),2),nodos(rutas_vuelta(i,~isnan(rutas_vuelta(i,:)
)),2),nodos(~isnan(rutas_vuelta(i,:)),3),nodos(rutas_vuelta(i,~isn
an(rutas_vuelta(i,:))),3)).';
end
```

## Anexo 6. Código de la presimulación

**CÓDIGO EN R**

```
set.seed(5843125)
      fecha=vector(mode = 'list',length = 5)
      maxSec=years(3)/seconds(1)
      for (te in 1:5) {
        fecha[[te]]=rexp(n = 1,rate = 1/ajuste_exp[bloq_hor=='b_1' & tipo_eme
      r==paste('t',te,sep='_'),lambda])
        pb=txtProgressBar(min = 0,max = 1,style = 3)

        while (last(fecha[[te]])<maxSec) {
          fecha[[te]]=c(fecha[[te]],last(fecha[[te]])+rexp(n = 1,rate = 1/aju
      ste_exp[bloq_hor==sec2BH(last(fecha[[te]])) & tipo_emer==paste('t',te,s
      ep='_'),lambda]))
          setTxtProgressBar(pb,last(fecha[[te]])/maxSec)
        }
        close(pb)
      }
      rm(pb,maxSec,te)
      nPartes=sum(unlist(lapply(fecha,length)))
      partes_nuevo=data.table(index=numeric(nPartes),
                              fecha_parte=numeric(nPartes),
                              bloq_hor=numeric(nPartes),
                              tip_vehi=character(nPartes),
                              TipEmergencia=character(nPartes),
                              lat=numeric(nPartes),
                              lon=numeric(nPartes),
                              nodo_id=numeric(nPartes),
                              cia=numeric(nPartes),
                              fecha_desp=numeric(nPartes),
                              fecha_sal=numeric(nPartes),
                              fecha_lleg=numeric(nPartes),
                              fecha_ret=numeric(nPartes),
                              fecha_ing=numeric(nPartes),
                              dt1=numeric(nPartes),
                              dt2=numeric(nPartes),
                              dt3=numeric(nPartes),
                              dt4=numeric(nPartes),
                              dt5=numeric(nPartes),
                              lat_cia=numeric(nPartes),
                              lon_cia=numeric(nPartes),
                              distIda=numeric(nPartes),
                              distRet=numeric(nPartes),
                              vel_ida=numeric(nPartes),
                              vel_ret=numeric(nPartes),
                              bloq_hor_ida=numeric(nPartes),
```

```
                                bloq_hor_ret=numeric(nPartes))


#Fecha de la emergencia
      partes_nuevo[,TipEmergencia:=rep(1:5,unlist(lapply(fecha,length)))]
      for (tipEmer in 1:5) {
        partes_nuevo[TipEmergencia==tipEmer,fecha_parte:=fecha[[tipEmer]]]
      }
      rm(tipEmer)
      partes_nuevo[,bloq_hor:=sec2BH(fecha_parte)]
      #Simular posición de emergencias
      set.seed(598453)
      pb=txtProgressBar(max = 5*42,style = 3)
      for (te in 1:5) {
        for (bh in 1:42) {
          distAct=spacial_dens[[paste('t',te,sep='_')]][[paste('b',bh,sep='_'
      )]]
          dens=melt(distAct$z)
          indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') & part
      es_nuevo$TipEmergencia==te)
          z_index=sample(x = 1:nrow(dens),size = length(indPartes),replace =
      T,prob = dens$value)
          bw_x=mean(diff(distAct$x))
          bw_y=mean(diff(distAct$y))
          partes_nuevo[indPartes,lon:=distAct$x[dens$Var1[z_index]]+runif(len
      gth(indPartes),-bw_x/2,bw_x/2)]
          partes_nuevo[indPartes,lat:=distAct$y[dens$Var2[z_index]]+runif(len
      gth(indPartes),-bw_y/2,bw_y/2)]
          setTxtProgressBar(pb,(te-1)*42+bh)
        }
      }
      close(pb)
      rm(pb,te,bh,distAct,dens,indPartes,z_index,bw_x,bw_y)


#Aproximar a nodos
      cl <- makeCluster(12)
      registerDoSNOW(cl)
      iterations <- nrow(partes_nuevo)
      pb <- txtProgressBar(max = iterations, style = 3)
      progress <- function(n) setTxtProgressBar(pb, n)
      opts <- list(progress = progress)
      result <- foreach(i = 1:iterations, .combine = c,
                        .options.snow = opts) %dopar%
                        {
                          return(nodos$ID[which.min(coord2m(nodos$lat,nodos$l
      on,partes_nuevo$lat[i],partes_nuevo$lon[i]))])
                        }
      close(pb)
      stopCluster(cl)
```

```
      partes_nuevo[,nodo_id:=result]
      rm(cl,iterations,pb,progress,opts,result)


#Número de vehiculos
      partes_nuevo[,nVehis:=NA_integer_]
      set.seed(46887)
      pb=txtProgressBar(max = 5*42,style = 3)
      for (te in 1:5) {
        for (bh in 1:42) {
          indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') & part
      es_nuevo$TipEmergencia==te)
          partes_nuevo[indPartes,nVehis:=sample(x = 1:88,size = length(indPar
      tes),replace = T,prob = nVehi[[te]][[bh]])]
          setTxtProgressBar(pb,(te-1)*42+bh)
        }
      }
      close(pb)
      rm(pb,te,bh,indPartes)


#Tipos de vehiculos
      partes_nuevo=partes_nuevo[rep(1:.N,nVehis),]
      partes_nuevo[,tip_vehi:=NA_character_]
      set.seed(46887)
      pb=txtProgressBar(max = 5*42,style = 3)
      for (te in 1:5) {
        for (bh in 1:42) {
          indPartes=which(partes_nuevo$bloq_hor==paste('b',bh,sep='_') & part
      es_nuevo$TipEmergencia==te)
          partes_nuevo[indPartes,tip_vehi:=names(disp_vehi)[sample(x = 1:13,s
      ize = length(indPartes),replace = T,prob = probVehi[[te]][[bh]])+1]]
          setTxtProgressBar(pb,(te-1)*42+bh)
        }
      }
      close(pb)
      rm(pb,te,bh,indPartes)


#Prioridad cias
      partes_nuevo[,prioridadCiaIndx:=split(dist_ida[match(nodo_id,nodos$ID),
      ],1:.N)]
      partes_nuevo[,prioridadCiaIndx:=lapply(prioridadCiaIndx,order)]


#dt1_2
      set.seed(9541)
      pb=txtProgressBar(max = 5*13,style = 3)
      tipVehiNames=names(disp_vehi)[-1]
      for (te in 1:5) {
        for (tv in 1:13) {
```

```
      indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]] & partes_
nuevo$TipEmergencia==te)
      partes_nuevo[indPartes,dt1:=rkernel(dt1_2_dens[[paste('t',te,sep='_
')]][[tipVehiNames[[tv]]]],N = length(indPartes),T)]
      setTxtProgressBar(pb,(te-1)*13+tv)
    }
  }
  close(pb)
  rm(pb,te,tv,indPartes)


#dt2
  set.seed(4567)
  pb=txtProgressBar(max = 13,style = 3)
  for (tv in 1:13) {
    indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]])
    partes_nuevo[indPartes,dt2:=rkernel(dt2_dens[[tipVehiNames[[tv]]]],N
= length(indPartes),T)]
    setTxtProgressBar(pb,tv)
  }
  close(pb)
  rm(pb,tv,indPartes)


#dt4
  set.seed(3246)
  pb=txtProgressBar(max = 5*13,style = 3)
  tipVehiNames=names(disp_vehi)[-1]
  for (te in 1:5) {
    for (tv in 1:13) {
      indPartes=which(partes_nuevo$tip_vehi==tipVehiNames[[tv]] & partes_
nuevo$TipEmergencia==te)
      partes_nuevo[indPartes,dt4:=rkernel(dt4_dens[[paste('t',te,sep='_')
]][[tipVehiNames[[tv]]]],N = length(indPartes),T)]
      setTxtProgressBar(pb,(te-1)*13+tv)
    }
  }
  close(pb)
  rm(pb,te,tv,indPartes)
```