

Una aplicación web, para asignación y ruteo de vehículos en caso de desastres

A web application for location and vehicle routing in disaster

Gustavo Gatica

Ph.D. Ciencias de la Ingeniería
Universidad Andrés Bello – Universidad de Santiago de Chile
Santiago de Chile, Chile
ggatica@unab.cl

Nicolás Venegas

Licenciado en Ciencias de la Informática
Universidad Andrés Bello (Chile)
Valparaiso, Chile
nico.venegas@uandresbello.edu

Rodrigo Linfati

Ph.D Investigación de Operaciones
Universidad del Bío-Bío (Chile)
rodrigo@linfati.cl

Carlos Contreras-Bolton

Ph.D (C) Universidad de Bologna
Universidad Santiago de Chile
Santiago de Chile, Chile
carlos.contrerasb@usach.cl

Omar Opazo

Ingeniero civil en Informática
Universidad Andrés Bello
Valparaiso, Chile
om.opazo@uandresbello.edu

John Willmer Escobar

Ph.D. Investigación de Operaciones
Universidad del Valle
Cali, Colombia
john.wilmer.escobar@correounivalle.edu.co

Resumen– Los desastres naturales, son eventos que exceden la capacidad de respuesta de una población y generan cuantiosas pérdidas, tanto económicas como humanas, con externalidades en muchos casos no cuantificadas en su totalidad. Los recursos necesarios para abastecer los centros de distribución son provistos tanto por proveedores privados como gubernamentales, deben ser asignados en virtud del daño del desastre. Luego viene la distribución desde los depósitos, hasta los distintos clientes o centros de distribución. Se presenta una aplicación web que asigna los superdepósitos, y luego establece el ruteo que han de seguir los vehículos para cubrir los centros de distribución, considerando diversas probabilidades de poblaciones que han de ser cubiertas. La aplicación es un marco de trabajo paramétrico a cualquier zona geográfica y escenarios, dado la integración existente con aplicaciones como Google Maps ®. Los tiempos computacionales son razonables, a nivel de arquitectura de software el producto es escalable y extensible. Además, cumple con un conjunto de buenas prácticas de calidad de software presentes en la ISO9126.

Palabras clave– Aplicación web, localización, logística humanitaria, ruteo de vehículos

Abstract– The natural disasters are events that exceed the capacity of covering of a population and generate large losses, both economic and humans, with externalities in many cases not quantified in their entirety. The resources needed to supply the distribution centers are provided both private and government must allocate providers, by the disaster damage. Then, the distribution is performed from the depots, to the different customers or distribution centers. It presents a web application that assigns the super depots, and then establishes the

routing that the vehicles must follow to cover the distribution centers, considering different probabilities of populations to be covered. The application is a parametric framework to any geographical area and scenarios, given the existing integration with applications such as Google Maps ®. Computational times are reasonable, and at the software architecture level the product is scalable and extensible. In addition, it complies with a set of good software quality practices present in ISO9126.

Keywords– Web application, Location, Humanitarian Logistics, Routing of vehicles.

1. INTRODUCCIÓN

Los desastres naturales son eventos que exceden toda capacidad de respuesta de cualquier estado y generan cuantiosas pérdidas, tanto económicas como humanas y, con un alto impacto social [1]. Los eventos generalmente no son predecibles, por ello se requiere de procedimientos y mecanismos que tengan la capacidad de minimizar los impactos y consecuencias del evento en cuestión [2]. Uno de los mecanismos empleados en caso de desastres, es establecer centros de agrupación de personas, para abastecer a la población de manera organizada en particular con suministros básicos, tales como, agua, alimentos perecederos, útiles de aseo, entre otros.

En este contexto, los recursos necesarios para abastecer a dichos centros son obtenidos, gene-

ralmente, desde los grandes supermercados, bodegas del Estado, puentes aéreos. Para comprensión de la problemática analizada, a estos centros se los denominará, *depósitos*. En tanto, los centros de agrupación de personas son posicionados sobre establecimientos de educación, gimnasios municipales, sedes comunitarias, iglesias, entre otros, a cada uno de estos puntos se les denomina *clientes*. Para el análisis que se presentará, se utiliza la experiencia empírica que proporcionó el terremoto del 27F en la ciudad de Concepción [3].

La distribución de los suministros desde los depósitos, hasta los distintos clientes, no puede dilatarse en tiempo ni tampoco en calidad y cantidad de productos. La carga de suministros se hace por medio de vehículos, los cuales deben ser enviados desde su punto de partida (depósitos) hacia sus puntos de llegada (clientes), se debe realizar eficientemente, es decir, cubriendo toda la demanda de los clientes, y a la vez optimizando las distancias recorridas, dado que un típico problema en caso de desastres es la disponibilidad de combustibles [4].

A la fecha, no se registran iniciativas que permitan integrar componentes gratuitos, tales como Google Maps, bibliotecas especializadas para ruteo de vehículo, y datos que pueden ser relevantes para casos de desastres. En el presente artículo, se desarrolla una aplicación web, que permite seleccionar un archivo con coordenadas pre-procesadas o seleccionando las ubicaciones mediante Google Maps®, depósitos, y el conjunto de clientes por visitar, donde para el caso de este último es posible modificar la demanda de manera paramétrica. La aplicación despliega gráficamente el ruteo que debe realizar una flota de vehículos homogéneos.

En la siguiente sección se presenta una definición del escenario de desastre abordado, para luego describir las características de la aplicación web, el resultado de las simulaciones, y un conjunto de conclusiones y trabajos futuros.

2. MATERIALES Y MÉTODOS

El escenario considerado como base para el desarrollo de la aplicación web analizado, se ha desarrollado con base en la experiencia que ha dejado el 27F en Chile [3], para ello se establece el escenario el cual consiste de establecer un

conjunto de depósitos, que corresponden a los grandes hipermercados y bodegas gubernamentales, que han de cubrir un conjunto de centro de personas (clientes). El escenario analizado es el que tiene una población superior a seis millones de habitantes [5]. Se realizan algunos supuestos que permiten abordar la problemática, entre los cuales destaca, un mapa con 416 clientes, considerando los locales de votación la información es obtenida de datos públicos disponibles [8]. Se establecen k depósitos para abastecer a los clientes del mapa, donde $5 \leq k \leq 15$; la distribución de la población entorno a un cliente (o centro de abastecimiento) es uniforme. Los depósitos cuentan con capacidad ilimitada; un vehículo puede recorrer x clientes mientras no se agote su carga; los vehículos son homogéneos; para el costo del trayecto solo se considera la distancia euclidiana. En la literatura lo anteriormente expuesto, corresponde a un clásico ruteo de vehículos con restricciones de capacidad y múltiples depósitos (MDVRP, por sus siglas en inglés de *Multi-Depot Vehicle routing problem*), considerando que se debe decidir respecto a qué depósitos se deben abrir o mantener cerrados, el problema ha sido abordado en la literatura con técnicas exactas y heurísticas.

El MDVRP es Np-Hard [6], y tiene por objeto determinar las rutas que puedan cubrir la demanda de todos los clientes con el mínimo costo de viaje, puede ser definido por un grafo completo no dirigido $G = (V, E)$, donde V y E , son los conjuntos de vértices y arcos, respectivamente. El conjunto V puede ser particionado en un subconjunto $I = \{1, 2, \dots, m\}$ de depósitos y un subconjunto $J = \{1, 2, \dots, n\}$, con demanda no negativa d_j y costos de tiempos de servicios δ_j . Cada depósito $i \in I$ tiene un tiempo de servicio $\delta_i = 0$, dado que para el caso del MDVRP pueden existir depósitos sin utilizar. Se dispone de un conjunto de k vehículos homogéneos, cada uno con capacidad Q , disponibles en cada depósito $i \in I$. Cada arco $(i, j) \in E$ tiene asociado un costo no negativo c_{ij} . Se debe además cumplir con cinco restricciones, como sigue:

- Cada ruta debe iniciar y terminar en el mismo depósito.
- Cada cliente debe ser visitado solo una vez, independiente de qué ruta sea asignado.

- El total de la demanda de cada ruta no debe exceder la capacidad del vehículo.
- El número de rutas asociadas a cada depósito no debe exceder la oferta de k vehículos.
- La duración total de cada ruta, no debe exceder un valor predefinido D .

2.1 Modelo matemático del problema

El problema puede ser definido mediante el siguiente modelo matemático:

2.1.1 Conjuntos

- D Potenciales centros de distribución (DC)
- J Clientes
- K Vehículos.

2.1.2 Parámetros

Se define un conjunto de parámetros, como sigue:

- N número de clientes,
- C_{ij} distancia entre dos nodos i y j , $i, j \in I \cup J$
- G_{ij} costo fijo de establecer el depósito $i \in I$
- F_k costo fijo de utilizar el vehículo $k \in K$
- V_i máxima capacidad de depósito $i \in I$
- d_j demanda de cliente $j \in J$
- Q_k capacidad del vehículo o ruta $k \in K$

2.1.3 Variables de decisión

Se definen cuatro variables de decisión:

$x_{ijk} = 1$, si el nodo i precede al nodo j en la ruta k ($i, j \in I \cup J$); 0 en otro caso.

$Y_i = 1$, si el depósito i es abierto; 0, en otro caso.

u_{ik} = Número de nodos visitados en cada ruta k hasta el nodo $i \forall i \in V; k \in K$

2.1.4 Modelo matemático

$$\begin{aligned} \text{Min } Z = & \sum_{i \in I} G_i y_i + \sum_{i \in I \cup J} \sum_{j \in I \cup J} \sum_{k \in K} C_{ij} x_{ijk} \\ & + \sum_{k \in K} F_k \sum_{i \in I} \sum_{j \in J} x_{ijk} \end{aligned} \quad (1)$$

Sujeto a:

$$\sum_{k \in K} \sum_{i \in I \cup J} x_{ijk} = 1, j \in J \quad (2)$$

$$\sum_{j \in J} d_j \sum_{i \in I \cup J} x_{ijk} \leq Q_k, \quad k \in K \quad (3)$$

$$N x_{ijk} \leq N - 1, \quad i \in I, j \in J, k \in K \quad (4)$$

$$\begin{aligned} u_{ik} + 1 & \leq u_{jk} + n(1 - x_{ijk}) \quad \forall i = 2, \dots, n; \\ & i \neq j; j = 2, \dots, n; k \in K \end{aligned} \quad (5)$$

$$\sum_{j \in I \cup J} x_{ijk} - \sum_{j \in I \cup J} x_{jik} = 0, \quad k \in K, i \in I \cup J \quad (6)$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1, \quad k \in K \quad (7)$$

$$\sum_{k \in K} \sum_{j \in J} d_j x_{ijk} - V_i y_i \leq 0, \quad i \in I \quad (8)$$

$$\sum_{u \in I \cup J} (x_{iuk} + x_{ujk}) \leq 1, \quad i \in I, j \in J, k \in K \quad (9)$$

$$x_{ijk} \text{ Binario}, \quad i \in I, j \in J, k \in K \quad (10)$$

$$y_i \text{ Binario}, \quad i \in I \quad (11)$$

$$u_{ik} \text{ Entera}, \quad i \in I, j \in J, k \in K \quad (12)$$

La función objetivo (1) minimiza la suma de los costos fijos de establecer los depósitos, costos de despacho y el costo por vehículo asignado, respectivamente. La ecuación (2) requiere que cada cliente debe estar asignado a una única ruta. Ecuaciones (3) están relacionadas con la capacidad del conjunto de vehículos. Ecuaciones (4) y (5) permiten considerar la eliminación de sub-tour. Restricciones de conservación de flujo se expresa en ecuaciones (6). Restricciones (7) garantizan que cada ruta es utilizada una única vez. La capacidad de los depósitos asignados se establece en restricciones (8). Restricciones (9) especifican que cada cliente puede ser asignado a un centro de distribución, si la ruta desde el centro de distribución pasa por dicho cliente. Ecuaciones (10), (11) y (12) se refieren a la naturaleza de las variables de decisión.

Escobar et al. [8] proponen un eficiente algoritmo tabú granular híbrido, mejorando lo propuesto en la literatura. Sin embargo, para este escenario, dado que las simulaciones han de ser por zonas geográficas, estas se pueden clusterizar y pre-procesar, según sea el requerimiento y la información

disponible en las bases de datos gubernamentales correspondientes [5].

La simulación es un recurso tanto didáctico como logístico, dado que permite adquirir enseñanzas a los participantes, Chile al tener una cultura sísmica, aprendizaje en casos de maremotos y erupciones volcánicas recurrentes, actualiza sus protocolos y se prepara para este tipo de situaciones, ya sea mediante simulacros o herramientas de simulación especializadas [9]. Sin embargo, existen lineamientos básicos a nivel paramericanos para estos casos [10] en Chile corresponde a la Oficina Nacional de Emergencias de Chile (ONEMI) adaptarlas a la realidad país. La ONEMI, en su sitio web tiene diversos planes, para distintos escenarios de desastres, y dispone de un Plan Nacional para la gestión de riesgos en caso de desastres [11]. Sin embargo, no se dispone de una aplicación web en el cual poder analizar diversos escenarios, considerando las experiencias de desastres anteriores. La experiencia en Chile, es declarar en algunos desastres en Estado de Excepción, donde las Fuerzas Armadas y de Orden deben resguardar la integridad de los hipermercados, cuyos productos fueron distribuidos a la comunidad, por efectivos de esta institución, por ello deben saber dónde asistir, con qué suministros y desde donde iniciar los circuitos de abastecimiento, lo anterior, se puede disponibilizar empleando herramientas gratuitas como la aplicación desarrollada.

La aplicación se desarrolló utilizando la metodología RAD, del acrónimo *Rapid Application Development* [19]. El proceso de captura de requerimientos fue mediante reuniones con expertos en desastre, y tomando como base lo sucedido el 27F.

3. TÉCNICA DE SOLUCIÓN

Se define una etapa previa al ruteo, con el fin de adaptar las entradas para el algoritmo seleccionado y así hacer posible la ejecución de este. La estrategia se denomina *Cluster first, route second* [12] y consta de dos etapas principales:

Cluster-first, se sabe que para una instancia en particular se pueden tener k depósitos $5 \leq k \leq 15$. Además, se tienen m clientes, que deben ser abastecidos. También se sabe que cada depósito y cliente, es una tupla, que entre otros datos, contiene la localización geográfica (longitud y latitud, con sus respectivas coordena-

nadas). La etapa *clúster* se encarga de pasar de un problema de k depósitos a k problemas de un depósito, bajo la norma de distancia euclidiana y utilizando el clásico algoritmo de *K-medias* [13]. La etapa se detalla en el siguiente algoritmo 1, el cual tiene un costo computacional de $O(k*m)$ [14].

La segunda etapa corresponde al ruteo de los vehículos, para ello se integra a la aplicación la biblioteca VRPH [15]. La biblioteca está desarrollada en lenguaje C++ y esta optimizada, obteniendo muy buenos tiempos computacionales para grandes instancias. La implementación es capaz de generar la ruta de cada vehículo disponible en el depósito, considerando las demandas de los distintos clientes y la capacidad (homogénea) de los vehículos. Finalmente, considera que el punto de partida de los vehículos (depósito) es de capacidad ilimitada. Sin embargo, la biblioteca no discrimina con respecto a la existencia de más de un depósito factible para el mismo cliente, por ello, justamente para el escenario analizado, se realiza un preprocesamiento en la primera etapa de la solución.

La solución a la problemática, en un entorno web, se muestra en el Fig. 1 en caso de catástrofe.

3.1 Arquitectura de la solución

La arquitectura definida para el desarrollo de la aplicación es híbrida [16], dado que en una de sus capas se puede visualizar un clásico estilo UNIX (*Pipe and filters*), para enviar la información por cada una de las tres capas de *software*. Desde esta capa se conecta al componente donde reside la biblioteca VRPH. Además, se establece un módulo para la interfaz de usuarios (UI), que permite la carga de los datos, el ajuste del contenido de la información al navegador que se esté utilizando, y el despliegue de los resultados.

La capa de *Pipe and filters* consiste de dos grandes componentes, el primero que permite procesar la data proveniente desde los archivos cargados por los usuarios, que contienen los *depósitos* y *clientes*, es decir, son preparados para ser procesados por el formato que solicita la biblioteca VRPH. El segundo, consiste en preparar los tipos de salida del procesamiento, que puede ser mediante el browser integrado con Google Maps®, o como archivos planos. Por qué se trabaja con ambas salidas, es solo ante la eventualidad

dad de que el servicio que proporciona Google ® no esté disponible.

A modo de auditoria, se desarrolla una capa de datos independiente DAL, acrónimo de Data acces layer, con el objeto de registrar cada una de las simulaciones realizadas, y así poder, en un trabajo futuro, mejorar la calidad de solución y/o agregar otras variables al modelo o modificar las rutas bajo demanda, con el objeto de contribuir a un sistema para desastres mucho más robusto.

La arquitectura a nivel de componentes funcionales se detalla en la Fig. 2, la cual responde en su integridad al escenario planteado en caso de desastres para el Gran Santiago, cabe destacar que las soluciones son proporcionadas en distancias euclidianas, dado que un factor siempre por considerar es la no disponibilidad de las calles, dado que pueden estar desbordadas, o simplemente cortadas como sucedió en la caso de la ciudad de Concepción para el 27F.

Fig 1. ALGORITMO EN ENTORNO WEB

Algoritmo Algoritmo propuesto de solución

Requerimiento de Entrada: Información dada por el usuario

Salida: Rutas de vehículos

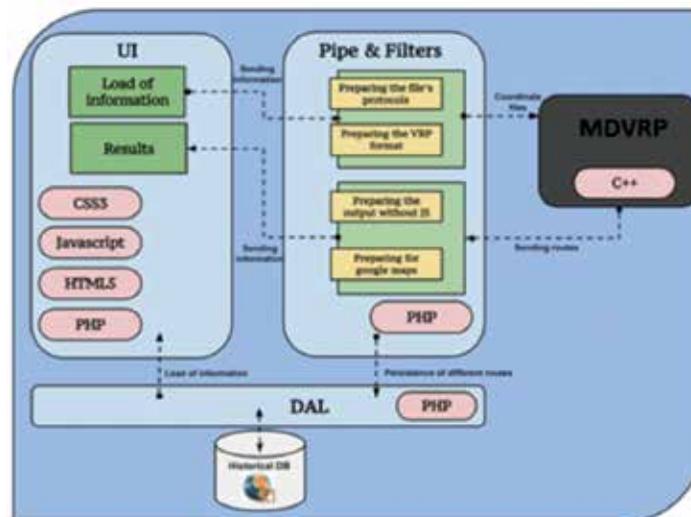
```

1: data=toTxt(data) {Primer filtro que solamente genera archivos de texto con información}
2: data=toVRP(data) {Segundofiltro que permite solamente nodos y coordenadas,(x_i,y_i)}
3: nodes = Cluster(datos[depósitos],datos[clientes])
4: min←∞
5: for i=0 to m do
6:   for j=0 to k do
7:     D=Distancia Euclideana (i,j)
8:     si D<min entonces
9:       min←D
10:      save_k←k
11:      save_m←m
12:     fin si
13:     nodes=Associate(save_k,save_m)
14:   end for
15: end for
16: for i=0 to k do
17:   Results=processVRP(i) {Este es procesado usando la libreria}
18: end for
19: return Results {EDD listo con todos los resultados de google maps}

```

Fuente: los autores.

Fig. 2. ARQUITECTURA FUNCIONAL



Fuente: los autores.

3.2 Interfaz de usuario

La interfaz de usuario se diseñó con el objeto de plasmar todo el contenido y funcionalidad en una única pantalla, para ello se utilizaron las normativas de usabilidad propuestas por Nielsen [17] y cumple con estándares de calidad, establecidos en la ISO 9126. La interfaz se puede visualizar en la Fig. 3, desde ahí se puede seleccionar los puntos precargados, los cuales fueron obtenidos con los datos públicos de los centros de votación disponibles a diciembre del 2014 [5]. También se puede cargar un conjunto de clientes y depósitos desde una planilla electrónica, el formato está predefinido, la plantilla está disponible en la misma sección. El procedimiento para realizar una ejecución está descrito en la misma sección, al pie de página.

Dentro de las secciones disponibles, se cuenta con una subsección de *Mantenimiento*, con el objeto de realizar acciones correctivas a los datos que están precargados en el sistema.

Fig. 3: PANTALLA DE INICIO



Fuente: autores.

3.3 Atributos de calidad

La aplicación desarrollada no tiene inconvenientes de operabilidad, dado que permite al usuario ejecutar la aplicación con cualquiera de los navegadores web que a la fecha se disputan el mercado. La aplicación es tolerante a fallas que presente el servicio de Google Maps, dado que

permite realizar los ruteos solo utilizando archivos y datos ingresados por usuarios. La arquitectura y posterior desarrollo del aplicativo, permite hacer cambios de los componentes sin afectar la lógica, es decir, se puede integrar sin mayor desafío de ingeniería a otro software que utilice mapas.

4. RESULTADOS DE SIMULACIÓN

El resumen de cada simulación, contiene el tiempo empleado, el número de depósitos seleccionados y los clientes cubiertos, como se presenta en la Fig. 4. Además, para cada depósito se indica su nombre, un listado de los clientes complementado con su ubicación, como se visualiza en la Fig. 5. Luego, se indican las rutas que son necesarias para cubrir la demanda total de los clientes asociados, como se puede visualizar en la Fig. 7.

Fig. 4. SIMULACIÓN 13 DEPÓSITOS Y 415 CLIENTES



Fuente: los autores.

Cabe destacar, que los clientes pueden ser visitados en más de una ocasión, ya que se debe cumplir con la demanda asociada a cada cliente.

Fig. 5. CLÚSTER QUE ABASTECE DEPÓSITO 1



Fuente: los autores.

Fig. 6. RUTAS PARA DEPÓSITO 1

189:	COLEGIO REPUBLICA DE GUATEMALA, GERMAN RIESCO 3315
180:	COLEGIO SANTA MARIA DE MAIPU , LOS PAJARITOS 4201
251:	ANEXO 2 LICEO MONSEÑOR ENRIQUE ALVEAR , LA ESTRELLA 205
264:	ESCUELA 1738 PUERTO FUTURO, LAGUNA SUR 8132
263:	ESCUELA 1602 ANTILHUE, ANCAHUAL 7457
005:	ESCUELA EJERCITO LIBERTADOR, ROSA ESTER RODRIGUEZ 6902
001:	ESCUELA D-259 CONDORES DE PLATA, SALOMON SACK 925
006:	ESCUELA REINA DE CHILE, 14 DE OCTUBRE 101
002:	ESCUELA D-262 LOTHAR KROMMER DRUGER, AVDA LAS TORRES 539
004:	ESCUELA D-279 JOAQUIN PRIETO VIAL, GOLFO DE MEXICO 361
Rutas	
001:	255, 260, 259, 253, 254, 164, 258, 257, 262, 179
002:	201, 200, 186, 195, 199, 182, 183
003:	198, 339, 333, 332, 43, 148, 150, 147, 3, 7
004:	197, 188, 191
005:	196, 190, 338, 341, 348, 337, 336, 342, 335, 343, 346, 347, 349, 334, 345, 194
006:	193, 192, 184, 187, 205, 203, 202, 204, 181, 185
007:	189, 180, 261, 264, 263, 5, 1, 6, 2, 4

Fuente: elaboración propia.

La información desplegada en las Figs. 5, 6 y 7, permiten en primera instancia visualizar gráficamente la ruta que deben seguir los vehículos para llevar su carga a cada centro de distribución, en tanto la Fig. 6, muestra enumerados los centros de distribución cubiertos por cada hipercentro, en caso de estar muy cercanos, se despliegan las rutas que ha de hacer como se muestra en la Fig. 6.

Las simulaciones realizadas permiten modelar diversos escenarios, como se puede observar, se incorpora el tiempo computacional empleado, por tanto, el cual es ínfimo, al ser una aplicación, puede disponerse de simulaciones concurrentes, las cuales no afectan al rendimiento computacional. Además, para minimizar uso de red, es factible prescindir de la integración con Google Maps, y trabajar solo con las rutas.

5. CONCLUSIONES Y TRABAJOS FUTUROS

En la actualidad y gracias al alto nivel desarrollado por las telecomunicaciones y de las redes computacionales, se ha llegado a un punto en el que los sistemas de *software* requieran de información online en cada uno de sus puntos, pero para países sudamericanos donde los censos tienen periodicidad de 10 años, es complejo. Pero se debe facilitar la integración con otros sistemas, como por ejemplo, capturar el número de señales de celulares que administra una celda, considerando que actualmente en Chile se tiene 1,7 celulares por persona [18], es factible obtener una buena distribución de la población.

La arquitectura de *software* definida y utilizada es portable, administrable, con multiplataformas

de acceso libre con componentes reutilizables, evolutiva, y de fácil análisis para futuras modificaciones y extensibilidad, además de ser 100% escalable. Esto se debe a que la Arquitectura de *software* definida está constituida por un conjunto de módulos y capas, que interactúan entre sí, que dispone de interfaces para comunicarse con otros subsistemas o servicios de otros proveedores.

Uno de los aspectos en los cuales se está trabajando, es la inclusión de un método capaz de balancear la carga entre los depósitos, posterior a la etapa de clusterización. Además, de considerar distintas distribuciones de probabilidades para los clientes.

REFERENCIAS

- [1] J.E. Vargas, *Políticas públicas para la reducción de la vulnerabilidad frente a los desastres naturales y socio-naturales*, Vol. 50. United Nations Publications, 2002.
- [2] *Manual para la evaluación del impacto socioeconómico y ambiental de los desastres*, Cepal, Santiago de Chile, Chile, 2003.
- [3] S. Barrientos, *Terremoto cauquenes 27 febrero 2010 Servicio Sismológico*. Universidad de Chile, May. 27 2010.
- [4] C. Daganzo, *Logistics systems analysis*, Springer, New York, 2005.
- [5] Gobierno de Chile. <http://datos.gob.cl>, 2014.
- [6] G. Laporte, Y. Nobert, and D. Arpin, *Optimal solutions to capacitated multidepot vehicle routing problems*, Vol. 44. Congressus Numerantium, Canada, 1984.
- [7] T. Wu, C. Low, and J. Wei Bai. "Heuristic solutions to multi-depot locationrouting problems". *Comput. Oper. Res.*, vol. 29, no. 10, pp. 1393-1415, Sep. 2002.
- [8] J. W. Escobar, R. Linfati, P. Toth, and M. G Baldoquin, "A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem". *Journal of Heuristics*, vol. 20, no. 5, pp. 483-509, 2014.
- [9] P. Aldunce and A. León, "Opportunities for improving disaster management in Chile: a case study". *Disaster Prevention and Management: An International Journal*, vol. 16, no. 1, pp. 33-41, 2007.
- [10] Guidelines for Developing Emergency Simulations and Drills. Organización Panamericana de la Salud, Washington, DC, 2010. Disponible en www.paho.org/disasters/index.php
- [11] Chile. ONEMI: Ministerio del Interior y Seguridad Pública. Disponible en <http://www.onemi.cl>

- [12] P. Toth and D. Vigo, *The vehicle routing problem*. Siam. Philadelphia, 2002.
- [13] N. Christofides and J.E. Beasley, "A tree search algorithm for the p-median problem". *European Journal of Operational Research*, vol. 10, no. 2, pp. 196-204, June 1982.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, et al., *Introduction to algorithms*, Vol. 2. MIT press Cambridge, 2001.
- [15] C. Groer, B. Golden, and E. Wasil, "A library of local search heuristics for the vehicle routing problem". *Mathematical Programming Computation*, vol. 2, no. 2, pp. 79-101, 2010.
- [16] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. vers, and R. Little. *Documenting software architectures: views and beyond*. Pearson Education, 2002.
- [17] J. Nielsen. *Designing web usability: The practice of simplicity*. New Riders Publishing, 1999.
- [18] Gobierno de Chile Subsecretaría de Telecomunicaciones. <http://www.subtel.gob.cl/>, 2015.
- [19] Martin, J. *Rapid application development*. Macmillan Publishing Co., Inc., 1991.