

UNIVERSIDAD DE CHILE
Facultad de Ciencias Físicas y Matemáticas
Departamento de ciencias de la computación
CC6908 Introducción al trabajo de título



*Visualización de estructuras
espaciales de desplazamiento a
partir de datos de transporte público*

18 de marzo de 2015

Profesor Guía
Marcela Munizaga

Profesor Co-guía
Benjamín Bustos

Memorista
Felipe A. Hernández G.
fhernand@dcc.uchile.cl
90977379

Índice

1. Introducción	1
2. Motivación	4
3. Objetivos	4
3.1. Objetivo General	4
3.2. Objetivos específicos	4
4. Metodología	5
5. Revisión de antecedentes	8
5.1. Análisis de bibliografía	8
5.1.1. Redes espaciales	8
5.1.2. Construcción y representación de la red	9
5.1.3. Análisis complejo de la red	9
5.1.3.1. Propiedades globales	9
5.1.3.2. Centralidades	10
5.1.3.3. Estructura de comunidad	11
5.1.4. Visualización de información	14
5.1.4.1. ¿Que es la visualización?	14
5.1.4.2. Tipo de visualizaciones	16
5.1.4.3. El proceso de visualización	18
5.1.4.4. El rol del usuario	20
5.2. Análisis de datos	22
5.2.1. Sistema tarifario	22
5.2.2. Estimación de bajadas	23
5.2.3. Descripción de los datos	24
5.3. Conclusiones	26
6. Estrategia de procesamiento	29
6.1. Herramientas a utilizar	29
6.2. Filtrado de datos	30
6.3. Construcción de la red	31
6.4. Trabajo pendiente	31
6.5. Solo punteos	32

7. Bibliografía	35
7.0.1. Instalación de Postgres	37
7.0.2. Creación de base de datos	37
7.0.3. Importación de los datos	46
7.0.4. Instalación Infomap	46
7.0.5. Instalación de R	47

Índice de tablas

1. Atributos a utilizar de la tabla ETAPAS	25
2. Atributos a utilizar de la tabla REDPARADAS	26
3. Atributos a utilizar de la tabla ESTACION_METRO	26

Índice de figuras

1. Metro subterráneo de Tokio	16
2. Mapa de GoogleMaps	17
3. Proceso de visualización.	18
4. Pipeline de visualización	19
5. Consulta para quitar etapas sin paradero de bajada.	30
6. Consulta para quitar atributos innecesarios de la tabla RED- PARADAS.	30
7. Consulta para quitar atributos innecesarios de la tabla RED- PARADAS.	31
8. Consulta para quitar atributos innecesarios de la tabla ESTA- CION_METRO.	31

1. Introducción

En los últimos años la utilización de tecnología en el transporte público ha ido en aumento debido a varios factores, mayor regulación, usuarios más exigentes, aumento en seguridad, etc. Lo anterior ha llevado al sistema público de transporte a implantar diversos dispositivos que permiten controlar los aspectos mas relevantes al momento de transportar una persona de un punto a otro.

Dentro de las tecnologías más usadas podemos nombrar AVL (Automatic Vehicle Location), que permite conocer la posición geográfica de un vehículo en todo momento con un margen de error bajo y los sistemas AFC (Automated Fare Collection) que automatizan el proceso de pago, en particular, nos interesa el basado en tarjetas de pago, que albergan un chip que permite mantener un saldo para que sea utilizado al abordar un bus, esto implica la existencia (paralelamente) de dispositivos asociados a los buses o paraderos¹ que permitan registrar el correcto descuento del valor asociado al pasaje, concepto que llamaremos validación.

Transantiago² es el sistema de transporte público de Santiago de Chile que implementa las tecnologías nombradas anteriormente, por lo que hoy en día se sabe que se realizan aproximadamente 6.000.000 de validaciones durante un día laboral³, lo que genera una cifra cercana a los 35.000.000 de transacciones a la semana (incluyendo sábado y domingo) con aproximadamente 3.000.000 de tarjetas de pago. Por otro lado, hay 80.000.000 de emisiones proveniente de la tecnología AVL del sistema. Al procesar estos datos en conjunto es posible identificar el paradero de origen, recorrido utilizado para desplazarse y el paradero de destino, este último requiere un procesamiento adicional basado en una metodología desarrollada por Munizaga y Palma (2012) [7] que logra una identificación en el 80 % de las validaciones.

La estructura espacial moderna de las ciudades ha sido formada, en gran medida, por avances en transporte y comunicaciones [1]. La forma en la cual se mueven los habitantes de una ciudad ha ido modificando la estructura de ésta, motivados por la transferencia de recursos como materiales, dinero, personas e información. Considerando una persona como un transportador de recursos de un área urbana a otra es que se identifican las siguientes

¹Lugar físico donde un bus de transporte público se detiene para que personas ingresen y/o desciendan a el.

²fue implementado a partir del año 2007.

³Lunes, martes, miércoles, jueves o viernes.

estructuras espaciales urbanas [4]:

- Centros de flujo: Se refiere a las áreas que sirven para conectar otro par de áreas para transferencia de personas. Funcionan como puentes espaciales entre distintas áreas.
- Centros: Se refiere a áreas que concentran personas. Pueden diferir de los Centros de flujo, pero a menudo, son lo mismo.
- Bordes: Se refiere a límites socioeconómicos generados a partir de la agrupación de paraderos que divide la ciudad en pequeños barrios que llamamos comunidades.

Lo anterior se enmarca en la necesidad de comunicar esta información de manera sintética dado que comprende una gran cantidad de datos y además abarcará otros campos de investigación, como lo es la arquitectura o planificación urbana, por lo que es necesario transmitir datos de forma clara y concisa.

De todo lo relatado podemos ver que hoy en día el sistema de transporte público de Santiago cuenta con una gran cantidad de datos pasivos por lo que existe una gran base de datos que mantiene un potencial de información que puede ayudar a identificar problemas de acceso a servicios básicos, además de tener la potencialidad de detectar otras necesidades. Sin embargo, las herramientas de procesamiento actuales no logran obtener toda la información que los datos proveen.

Según lo anterior, el problema que se busca resolver en esta memoria es interesante de abordar debido a que ayudará a entender la estructura espacial de los viajes realizados por la población. Esto ayudará a:

- Mejorar las posibilidades de realizar actividades en el entorno de la zona de residencia.
- Disminución de la demanda en los Centros de flujo.
- Disminución en el tiempo requerido para trasladarse hasta el punto de interés para una comunidad determinada.
- Mejorar las condiciones de viaje de grupos vulnerables.

Esta memoria creará las estructuras espaciales mencionadas anteriormente a partir de los datos de transporte público de Santiago de Chile y diseñará una aplicación para que puedan ser visualizadas, de manera de comunicar información existente en los millones de datos provistos y que las herramientas actuales de procesamiento no abarcan.

2. Motivación

Como hemos dado a conocer, existe una gran fuente de datos con mucha información pero que actualmente encuentra sus dificultades en el procesamiento y la forma en que puede ser comunicada. Por lo que una solución a este problema puede abrir las puertas a nuevas preguntas y según esto, nuevas investigaciones.

También es interesante académicamente debido a la masividad de los datos, ya que se deberá implementar una estrategia de procesamiento que permita manejar millones de registros y además realizar análisis sobre estos que permitan comunicar información por medio de la visualización.

3. Objetivos

3.1. Objetivo General

“Diseñar una herramienta que permita identificar y visualizar estructuras espaciales de movimiento en la ciudad de Santiago de Chile utilizando datos pasivos y masivos de transporte público.”

3.2. Objetivos específicos

1. Construir modelo de red para la ciudad de Santiago de Chile.
2. Identificar patrones de viaje, centros y centros de flujo.
3. Desarrollar una herramienta que permita visualizar las estructuras espaciales.

4. Metodología

Esta metodología está basada en una investigación publicada en la *International Journal of Geographical Information Science* [4], por lo que los procedimientos ya han sido probados en otro contexto, reduciendo de esta forma posibles inconvenientes que puedan ocurrir a lo largo del desarrollo de esta memoria.

Los datos a utilizar se han definido como los producidos en una semana de calendario (lunes a domingo). Estos ya se encuentran procesados según la metodología diseñada por Munizaga y Palma (2012)[7], por lo que los datos corresponden a una tabla de una base de datos PostgreSQL llamada *tabla_de_etapas* donde cada fila representa una etapa de un viaje⁴. Según lo anterior la cantidad de datos a utilizar es de aproximadamente 35.000.000, que corresponde a la cantidad de etapas realizadas por el 80 % de las transacciones del sistema AFC⁵.

Dado lo anterior, el desarrollo de esta memoria considera la siguiente metodología de trabajo:

1. Investigación bibliográfica

Se está realizando una recopilación y redacción de las ideas y estrategias más relevantes que aporten y justifiquen la base teórica de esta memoria.

2. Estudio de los datos.

Se realizará un estudio de los datos existentes para comprender concretamente las bases de datos requeridas.

3. Definición de estrategia de pre-procesamiento de datos.

Se investigará sobre las estrategias de pre-procesamiento y elegirá la que mejor se adapte en base al estudio realizado en el ítem anterior. Dentro de esta etapa se llevará a cabo la normalización y selección de los datos para realizar los análisis.

4. Construcción de la red de nodos.

En esta etapa se realizará la construcción de un grafo dirigido con nodos a partir de los estudiados.

⁴un viaje puede tener una o más etapas.

⁵Automatic Fare Collection

5. Análisis de la red

a) Definición de propiedades básicas.

Aquí asociaremos un atributo de las estructuras urbanas a cada propiedad matemática de un grafo a partir de las ideas obtenidas de la investigación bibliográfica.

b) Definición de centralidades.

1) Centro de flujo.

Se define el concepto de Centro de flujo en un grafo (*centros de intermediación*) y se propone una fórmula para medirlo.

2) Centro

Aquí estudiaremos y definiremos la estrategia para detectar centros de la ciudad ocupando el algoritmo *PageRank*.

c) Estructura de comunidad.

Para la detección de estructuras de comunidad se utilizará el software *infomap*.

6. Análisis espacial

a) Interpolación espacial.

Lo relevante de esta etapa es relacionar una zona geográfica a un paradero de bus de manera de poder particionar la ciudad.

b) Cálculo estadístico.

En esta etapa se realizará la asociación de las comunidades detectadas a las áreas geográficas establecidas en la interpolación espacial. El principal desafío aquí es corregir aquellas coordenadas que pertenecen a una comunidad en la red pque no están geográficamente adyacentes al grupo principal que define la comunidad.

7. Análisis de los resultados.

Se estudiarán los resultados obtenidos.

8. Definir visualizaciones y nivel de interactividad de cada una.

A partir del punto anterior se definirán las visualizaciones a realizar y las posibles interacciones que puedan haber en cada una de ellas.

9. Diseño de aplicación de visualización.

Se desarrollará una aplicación que permita ver cada una de las implementaciones definidas en el punto anterior.

Este trabajo será realizado a lo largo de 2 semestres (2014-2 y 2015-1) por lo que se dividirá de la siguiente forma:

- Semestre 2014-2

1. Investigación bibliográfica
2. Estudio de los datos
3. Definición de estrategia de pre-procesamiento de datos

- Semestre 2015-1

4. Construcción de la red de nodos
5. Análisis de la red
6. Análisis espacial
7. Análisis de los resultados
8. Definir visualizaciones y nivel de interactividad de cada una
9. Diseño de aplicación de visualización

Es importante decir que el punto 3 se espera abordarlo de manera parcial, realizando un acercamiento durante este período para luego finalizarlo previo inicio del segundo y así poder lograr el desarrollo del resto.

5. Revisión de antecedentes

5.1. Análisis de bibliografía

Esta memoria se basa principalmente en la metodología propuesta por Cheng Zhong et al. (2014) [4], la cual provee un método cuantitativo para la detección de **Centros de flujo**, **Centros** y **Bordes** pudiendo identificar estructuras urbanas a partir de datos pasivos de transporte público. Dentro de este mismo documento se establece una vinculación entre los centros de flujo, centros y bordes con fenómenos urbanos reales dado que utiliza un grafo generado a partir de datos obtenidos del comportamiento de la gente. Además permite aplicar nuevas técnicas para detección de bordes basadas en otras metodologías que puedan aparecer en el futuro, permitiendo la comparación entre ellas. Por lo tanto, la metodología allí expuesta es utilizada en esta memoria para la generación de las mismas estructuras pero para ser analizada con los datos locales de transporte público.

5.1.1. Redes espaciales

Una red espacial se entiende como un grafo cuyos vértices y arcos representan objetos geométricos del mundo real. Los nodos tienen una posición relativa a un sistema de referencia específico y los arcos expresan la forma física en que interactúan entre ellos, entendiéndose esto último como la forma en que se puede llegar físicamente de uno a otro. El termino “complejo” se anexa al término de red espacial cuando el grafo de esta red se caracteriza por tener propiedades estructurales que se relacionan con la accesibilidad de un nodo y su aporte dentro de la red. Estas propiedades tienen la particularidad de estar presentes en muchos problemas reales (Doursat 2005) [5].

Hace algunos años atrás los análisis espaciales urbanos se limitaban a utilizar el diseño de las calles en términos de su topología urbana (Cardillo et al. 2006)[3], lo que tiene la limitante de no considerar la accesibilidad asociada a una calle como una característica dependiente de los movimientos humanos existentes. Además, este tipo de análisis tiende a ignorar los flujos urbanos y a justificar espacios y su forma en función de las propiedades de la red.

En los últimos años, los estudios sobre estas redes (Soh et al. 2010) [14] comenzaron a incorporar medidas de peso que reflejan los datos de movimientos urbanos como flujos sobre la red pero concentrado en el sistema de trási-

to, no sobre los espacios urbanos asociados a estos .

Además de los datos obtenidos a partir del transporte público han existido investigaciones basadas en otras fuentes de datos, como lo son los AVL basado en GPS (Rinzivillo et al. 2012)[11] o conjuntos de datos telefónicos (Ratti et al. 2010)[10]. En particular, el artículo de Cheng Zhong et al. (2014) utiliza los datos generados a partir del uso de tarjetas inteligentes del transporte público de Singapur.

5.1.2. Construcción y representación de la red

Formalmente definimos un grafo dirigido con pesos como $G = (N, L, W)$ que representa todas las etapas de viajes realizadas. N denota los paraderos existentes, a éste se asocia el área o sector donde está ubicado, el conjunto L denota los traslados entre dos paraderos o áreas, por lo que L corresponde a un conjunto de pares ordenados de N , y el conjunto W denota el volumen (cantidad) de traslados entre dos paraderos. Según lo anterior, N son los nodos del grafo, L representa los arcos y W denota los pesos de cada arco en L .

Por otro lado, Munizaga y Palma (2012) [7] (explicado en la sección 5.2) proponen una metodología para crear una *matriz Origen-Destino*, generando la oportunidad de modelar la construcción teórica descrita en el párrafo anterior en sistemas donde solo se valida en un sentido, como lo es en el caso de Santiago de Chile.

5.1.3. Análisis complejo de la red

Este análisis se abarca desde 3 perspectivas: propiedades globales, información local asociada a *centros* y *centros de flujo*, y en la detección de comunidades.

5.1.3.1. Propiedades globales Las propiedades topológicas de un grafo pueden proveer importante información sobre las interacciones espaciales que se producen en el modelo real, según esto, se define lo siguiente

- El número N de nodos indica cuantos paraderos o áreas son accesibles, y el número de arcos J indica cuantos paraderos o áreas están directamente conectadas.

- El **grado** de cada nodo en la red indica cuantos paraderos o áreas están directamente conectadas desde una en particular, aquí podemos diferenciar entre **grado de salida** (cantidad de nodos que tienen traslados cuyo origen es ese paradero) y **grado de entrada** (cantidad de nodos que tienen como destino ese paradero).
- El **peso** de cada arco indica la intensidad (volumen) de traslados desde un paradero a otro.
- La **ruta más corta** se refiere al camino mínimo posible de un área a otra.
- “**clustering centrality**” es un índice que mide cuán cohesionados o cercanos están los nodos a otro en términos de su accesibilidad para compartir vecinos.
- **Centralidad de cercanía** es un índice que evalúa cuan rápido viaja la información en un grafo. Se calcula para un nodo i como $\frac{1}{lejanía(i)}$, donde $lejanía(i) = \frac{1}{|V|-1} \sum_{j \neq i} \delta_{ij}$. Donde δ_{ij} es la distancia más corta de i a j .

Estas propiedades permiten descubrir los niveles de actividad que mantiene cada área de una ciudad y su participación en los flujos que se realizan.

5.1.3.2. Centralidades A las propiedades generales agregamos 2 tipos de centralidad adicionales, una es la *centralidad de intermediación*, la que se usará para definir los **centros de flujo** y el segundo es el *PageRank* que mide la accesibilidad en la red tomando en cuenta todos los vínculos, directos e indirectos, sus pesos y dirección. Este último indicador será útil para medir el grado en que cada nodo es un **centro**.

La centralidad de intermediación es un indicador que mide cuán bien conectada está un área o paradero, formalmente se define para un nodo k como el número de caminos más cortos que conecta dos áreas i y j en el grafo que pasan a través del nodo k , y se define como

$$C_{intermediacion}(k) = \sum_{ij} \frac{\delta_{ij}(k)}{\delta_{ij}}$$

donde $\delta_{ij}(k)$ es el número de caminos más cortos entre i y j que pasan por k , mientras que δ_{ij} es el número total de caminos más cortos entre i y j . Notar que i y j son distintos de k .

Por otro lado, *PageRank* mide el rol de un nodo o área local en atraer flujos desde todos los nodos de la red. Esta medida puede ser vista como una representación genérica de las probabilidades de un peatón cualquiera de visitar un nodo cualquiera, en este sentido está relacionado con procesos de Markov de primer orden (Brin y Page 1998)[13], que es la base de muchos procesos de interacción social, en este contexto fue originalmente usado para extraer información acerca de las estructuras de los vínculos de Internet (Rosvall y Bergstrom 2008)[12], muy similar al *ranking de páginas de Google*. Según lo anterior se define la probabilidad r_j de visitar el nodo j como:

$$r_j = [(1 - \rho)/N] + \rho \sum_i r_i p_{ij}$$

Donde $1 - \rho$ puede ser visto como la probabilidad de que un caminante decida quedarse en el nodo j , y p_{ij} como la probabilidad de escoger ir al nodo j dado que estoy en el nodo i , este valor es proporcional al peso del arco de i a j , en resumen

$$p_{ij} = w_{ij} / \sum_k w_{ik}, \text{ y } \sum_j p_{ij} = 1$$

El parámetro ρ es conocido como *factor de amortiguación* y toma valores entre 0 y 1, en el estudio de Chen Zhong et al. (2014) fue fijado en 0,85. Si $\rho = 1$ entonces todos los nodos tienen una probabilidad positiva y luego, la matriz $\{p_{ij}\}$ tiene que estar fuertemente conectada.

5.1.3.3. Estructura de comunidad Los bordes a identificar sobre la superficie a analizar sirven para particionar la estructura espacial y así crear pequeños vecindarios a partir de ésta que denominamos comunidades. Estos son obtenidos a partir de la detección de una *estructura de comunidad*, que se refiere a una **propiedad de un grafo que permite agrupar nodos de éste que están densamente conectados entre ellos en comparación con el resto de nodos del grafo**. Según lo anterior, los bordes son generados a partir de un descriptor de bordes que particiona la red en dos niveles donde los nodos forman módulos que llamamos comunidades y la división entre estos que llamamos bordes. Existen varias formas de generar

comunidades pero una condición necesaria para este trabajo es la consideración de las variables de **densidad** y **flujo de interacciones** al momento de crear éstas. Lo anterior basado en que estas dos variables deben ser mas fuertes dentro de una comunidad y que el volumen que está dentro de cada comunidad es mayor en comparación con el resto de la red.

Para la generación de las comunidades se utiliza el framework *mapEquation* basado en un procedimiento llamado *infomap* desarrollado por Rosvall y Bergstrom el 2008 [12]. Lo anterior se justifica por Lancichinetti y Fortunato (2009)[6], donde concluyen que es uno de los algoritmos que ha mostrado mejor rendimiento para la generación de comunidades y uno de los pocos adecuados para redes con peso y dirección. Otra característica relevante del algoritmo *infomap* es que no solo considera la relación entre pares de nodos sino que también toma en cuenta los flujos presentes entre estos.

Para llevar a cabo lo anterior se utilizan flujos probabilísticos creados a partir de generaciones aleatorias que simulan recorridos sobre el grafo y que asignan también probabilidades a cada nodo de ser visitado aleatoriamente (utilizando el algoritmo *PageRank*), con el objetivo de modelar los comportamientos de flujo de un sistema real.

En resumen, el algoritmo divide los nodos del grafo en módulos que son altamente estructurados, lo que implica que la entropía del grafo particionado es mínima (Rosvall y Bergstrom, 2008)[12]. Esta entropía total del sistema está dividida en la entropía de moverse entre los módulos más la entropía de moverse dentro de un módulo, las proporciones están relacionadas a la probabilidad de ocurrencia de cada uno. Por lo anterior, Rosvall y Bergstrom (2008) definen esta entropía como:

$$\left. \begin{aligned} Lg(M) &= H(P) + \sum_{i=1}^m P_i H(p)_i \\ &= -p \sum_{i=1}^m P_i \log P_i - \sum_{i=1}^m P_i \sum_{k=1}^{M_i} \frac{P_k}{P_i} \log \frac{P_k}{P_i} \end{aligned} \right\}, P_i = \sum_k p_k$$

donde P_i es la probabilidad de ser visitada la comunidad i , p es la probabilidad que un caminante cualquiera de cambiarse de modulo y P_k es la probabilidad de visitar el nodo k . Además M_i es la cantidad de nodos que contiene la comunidad i y la división por P_i que afecta a los P_k cumple la función de normalizar.

La forma en que trabaja es primeramente asociando cada nodo al módulo que pertenece, donde en cada paso, se identifica que nodo se puede agregar a

que módulo tal que la entropía general decrezca. Este proceso continua hasta que no se pueda reducir más la entropía, asegurando que con esa configuración se obtiene la partición más estructurada.

Es importante decir que M_i es un módulo que contiene un conjunto de $k \in M_i$ nodos y que llega a ser estable (sin alteraciones) una vez que se obtiene la entropía mínima (Rosvall y Bergstrom, 2008). Luego, estas comunidades son mapeadas a sus respectivas ubicaciones geográficas.

Con lo anterior ya hecho, lo que restó por hacer fue transformar los conjuntos de puntos discretos obtenidos por *PageRank* en regiones que particionan el espacio geográfico, para esto se realizó una interpolación espacial que considera el siguiente supuesto:

“Cada persona escoge el paradero de bus/metro más cercano a la posición en la que se encuentra”

Según lo anterior, Chen Zhong et al. (2014) propone aplicar una interpolación a cada zona geográfica cercana a un paradero de bus/metro. La variante de interpolación escogida corresponde a la *Inverse Distance Weighting (IDW)* lo que hace disminuir la influencia de un paradero de bus en función de la distancia. Estos pesos son definidos de la siguiente forma

$$W_i(x, y) = 1/d_{ij}(x, y)^\lambda$$

Donde $W_i(x, y)$ es el peso del paradero i en las coordenadas (x, y) que son los puntos vecinos más cercanos a j y $d_{ij}(x, y)$ es la distancia a la coordenada (x, y) desde el paradero i hacia el paradero vecino más cercano j .

Una observación que se desprende de la fórmula es que los pesos están normalizados tal que su suma sea 1, es decir, $\sum_{\forall x, y} W_i(x, y) = 1$ y λ es un parámetro arbitrario que en este caso es 2, lo que implica que sigue la ley del inverso al cuadrado.

Por último cada coordenada espacial fue asignada a una única comunidad, esto lo logran utilizando un *resumen estadístico*⁶. El principal problema aquí fue tratar aquellas coordenadas que pertenecen a una comunidad en la red pero que no están geográficamente adyacentes al grupo principal que define la comunidad, esto se origina porque el algoritmo de detección de comunidades no está restringido a lograr áreas geográficamente contiguas.

⁶Se entiende como la información dada por una rápida y simple descripción de los datos como la media, mediana, moda, rango y desviación estándar.

Aunque la investigación reporta que esto no ocurre muy a menudo, y que cuando sucede, es en los límites de las áreas que definen las comunidades y se da principalmente porque las personas que viven en esas áreas tienen diferentes preferencias de viajes.

Para solucionar el problema planteado se cuenta el número de puntos en las comunidades en conflicto y se computa el algoritmo *PageRank*, esto provoca que las coordenadas en disputa sean asignadas a su comunidad más cercana geográficamente. En la práctica lo que se ocurre es un desplazamiento de los límites entre comunidades, de esta forma se obtiene una partición geográfica que abarca todo el espacio.

5.1.4. Visualización de información

En esta sección serán descritos algunos aspectos relevantes de la visualización de información con el fin de justificar teóricamente preguntas como: ¿qué es la visualización? ¿qué colores utilizar? ¿qué tipo de visualización mostrar? ¿cuál es el objetivo de la visualización que estoy creando? entre otras.

Toda la información necesaria fue obtenida del libro *Interactive Data Visualization: Foundations, Techniques, and Applications* [8].

5.1.4.1. ¿Qué es la visualización?

El concepto de visualización se define como la *comunicación de información usando representaciones gráficas*. El beneficio de utilizar imágenes está dado por la riqueza de información que puede contener y el tiempo que requiere para ser procesada en vez de información en páginas con texto, esto ocurre debido a que la interpretación de imágenes es realizada en **paralelo** dentro del sistema perceptual mientras que la lectura es un proceso **secuencial**. Otra de las ventajas que tiene la visualización es su independencia del lenguaje, siendo más eficaz que un texto para un grupo de personas que hablan diferentes idiomas.

La importancia de esta forma de comunicación radica en que los seres humanos son seres visuales que usan la visión como uno de los sentidos claves para entender la información, según esto es que se justifica el hecho de que sea ocupada en muchos lugares en la toma de decisiones, y dado el incremento de datos presentes hoy en día en cada área del conocimiento, es que hay una

creciente necesidad por herramientas y técnicas que ayuden a convertirlos en información útil.

permitiendo proveer información detallada sobre diversos puntos de interés. En la figura 2 se observa un mapa con una ruta dirigida entre dos puntos, indicando calles, sentido de estas, intersecciones, y alternativas de recorrido (medibles y comparables).

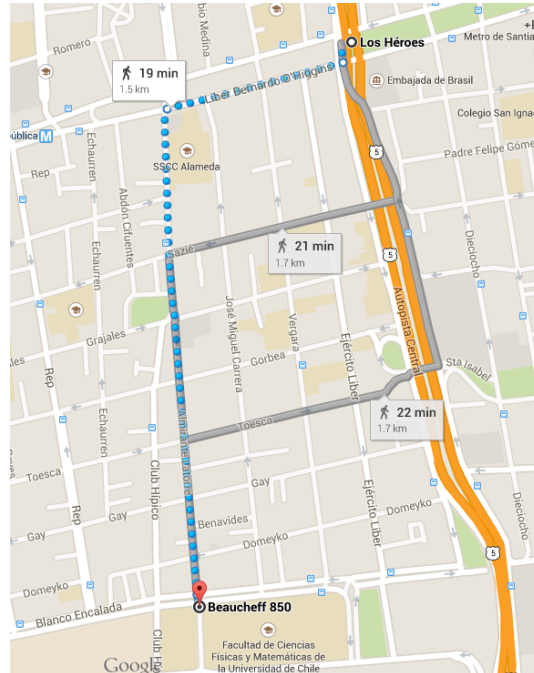


Figura 2: Mapa que indica posibles rutas a seguir (caminando) desde beaucheff 850 a la estación de metro Los Heroes. (obtenida desde *maps.google.cl*)

Lo anterior sugiere que las imágenes distorsionadas tienden a ser imprecisas y dependen de quien las observa.

Otra manera de diferenciar la visualización es según el nivel de interacción que ésta presenta. Según esto tenemos

Visualización estática no permite interacción, muy parecido a una imagen.

Visualización dinámica permite interacción, aquí el usuario puede consultar en la visualización y en consecuencia, interactuar con la aplicación de visualización en lugar de usar un menú, pudiendo llegar incluso a tener aplicaciones que son totalmente dirigidas a través de su visualización.

La visualización enfatiza muchos aspectos de otras disciplinas, abarcando áreas como la computación gráfica, psicología perceptual, interacción humano-computador, base de datos, estadística, y minería de datos, entre otras.

Hoy en día existen dos corrientes de visualización, una científica y otra de información. Ambas proveen representaciones de datos, pero la diferencia más característica es que estos datos suelen ser a menudo diferentes.

5.1.4.3. El proceso de visualización

Los datos utilizados para una visualización pueden ser simples o complejos en su estructura, y quien observa podría tener distintos objetivos sobre esos datos (buscar algo interesante como anomalías, tendencias, confirmar una hipótesis, mostrar resultados de análisis a una audiencia, etc). Por esto, uno de los primeros pasos para llevar a cabo una visualización es realizar un mapeo desde los datos a la vista (ver figura 3), y no hay una única forma de hacerlo.

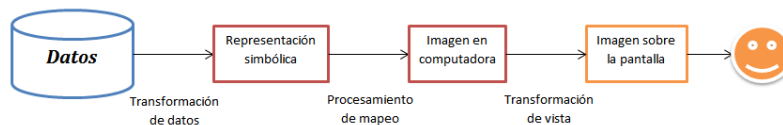


Figura 3: Proceso de visualización.

En la figura 3 se observa un proceso de visualización genérico que describe a grandes rasgos las etapas presentes. Según esto las etapas son:

Transformación de datos En esta etapa se realizan una o más operaciones⁷ (limpieza, correcciones, etc) que convierten los datos desde la fuente de datos en una representación simbólica en la computadora.

Procesamiento de mapeo Cada representación simbólica es asociada a un objeto gráfico cuyas características suelen representar atributos de cada elemento obtenido en el paso anterior. El conjunto de estos objetos producen una imagen en un computador.

⁷Todo lo relacionado con el pre-procesamiento de datos

Transformación de vista Cada imagen puede ser transformada según requerimientos que tenga el usuario o si hay un grado de interacción, éste tiene la posibilidad de hacerlo en el momento que lo requiera.

Es importante decir que no hay una fórmula que garantice la efectividad de una visualización. Diferentes usuarios, con diferentes formaciones, habilidades perceptuales y preferencias, tendrán diferentes opiniones sobre una visualización. La tarea que deban realizar los usuarios afectará también la utilidad de la visualización.

La visualización es a menudo una parte de un proceso más grande como análisis de datos, descubrimiento de conocimiento o análisis visual.

El proceso de comenzar con datos y generar una imagen es descrito como *pipeline*, que se define como una secuencia de etapas que puede ser estudiada independientemente en términos de algoritmos, estructura de datos y sistema de coordenadas.

A continuación se muestra una descripción del pipeline de visualización. Las etapas son descritas en el mismo orden en que son ejecutadas durante el proceso (ver figura 4).

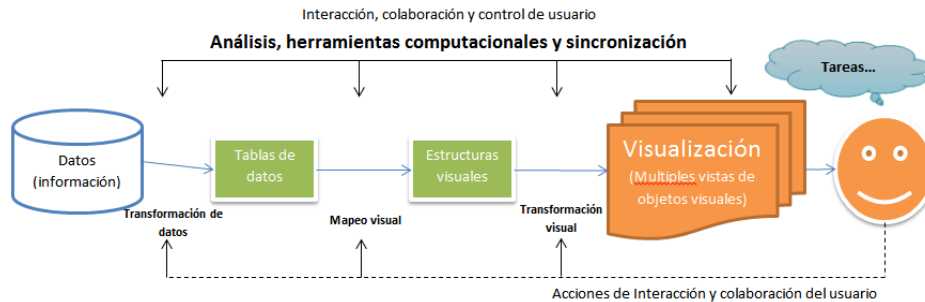


Figura 4: Ejemplo de pipeline de visualización. Muestra la idea esencial de un pipeline, que es transformar los datos en una representación interna dentro del computador para luego usar un paradigma visual para desplegarlo sobre una pantalla.

Modelado de datos

Si están en un archivo o en una base de datos, tienen que estar estructurados. Esta estructura se evalúa en base a dos conceptos, sintaxis, relacionada con el significado de su representación y semántica, visto como la relación existente entre registros. Además se realiza una limpieza de los datos, aquí se pueden utilizar varias estrategias (eliminar registros malos, valor sentinela, reemplazar por valor más cercano,

etc) para luego identificar cada uno de los tipos de datos existentes (Ordinal, Nominal). Por último, si hay datos faltantes puede recurrirse a la interpolación.

Selección de datos

Se relaciona con la identificación del subconjunto de datos que serán potencialmente visualizados. Esto puede ocurrir totalmente bajo el control del usuario o a través de un algoritmo automatizado.

De los datos a mapeo visual

Este es lo más importante dentro del pipeline, la realización del mapeo de los valores de los datos a entidades gráficas con sus respectivos atributos. Esta etapa usualmente involucra un procesamiento previo de los datos a ser mapeados como escalamiento, desplazamiento, filtrado, interpolación, entre otros. Esta etapa es muy dependiente de la librería subyacente en la visualización.

Configuración de parámetros de la escena

Como en las gráficas tradicionales, el usuario tiene que especificar muchos atributos de la visualización que son relativamente independiente de los datos. Esto se refiere a configurar parámetros como los colores (para diferentes dominios, ciertos colores tienen significados claramente definidos), o especificación de luz si se tratase de una visualización 3D, etc.

Renderizado o generación de la visualización

La proyección específica o el renderizado a ser utilizado para los objetos a visualizar depende del mapeo que se ha utilizado. En este aspecto, la mayoría de las visualizaciones solo requieren dibujar líneas y polígonos uniformemente pintados. Junto a mostrar los datos en sí mismo, muchas también incluyen información suplementaria para facilitar la interpretación, como ejes, anotaciones, entre otros.

5.1.4.4. El rol del usuario

El usuario puede estar envuelto en la mayoría de las etapas del pipeline de visualización. Un aspecto importante desde este punto de vista es la intensidad del usuario, según esto, es útil categorizar la visualización basados en los propósitos que ésta tendrá. Según esto surgen las siguientes categorías:

Exploración

El usuario posee un conjunto de datos y quiere examinarlos para averiguar su contenido y/o si una característica particular (o conjunto de estas) está presente o ausente.

Confirmación

El usuario ha determinado o planteado una hipótesis relacionada con una característica de los datos y quiere usar la visualización para verificarla.

Presentación

El usuario está intentando transmitir algún concepto o conjunto de hechos a una audiencia.

5.2. Análisis de datos

A continuación se detalla la forma en que funciona el sistema de transporte público que origina los datos, como se estima el proceso de bajada y por último los detalles de las tablas obtenidas posterior a la estimación. Tanto la metodología como los supuestos aquí explicados son expuestos y propuestos por Munizaga y Palma (2012) [7].

5.2.1. Sistema tarifario

En Santiago de Chile, el sistema AFC utilizado corresponde a las tarjetas de pago, donde en buses es el único método disponible y en metro es el más utilizado.

El sistema de pago en Transantiago es tal que cada pasajero paga una tarifa cuando accede al sistema, que permite a él o ella hacer tres transbordos dentro de las dos horas siguientes al pago. La estructura de pago es diferente entre el metro y buses. En buses, el único sistema de pago es mediante la tarjeta de pago (llamada comercialmente tarjeta bip!), mientras que en metro, es posible comprar un ticket o usar la tarjeta bip!, sin embargo el porcentaje de usuarios que compra el ticket es de aproximadamente 3 %.

El sistema se caracteriza por tener cerca de 300 rutas de buses, 6000 buses disponibles agrupados en 6 operadores⁸, aproximadamente 10.000 paraderos y una cifra que bordea los 150 kilómetros de rieles para el metro.

Dada la alta demanda que ha experimentado, se crearon 150 zonas físicas llamadas “zona paga” que están equipadas con sistema de pago (validadores) de vehículo donde el pasajero paga cuando entra a la estación, lo cual incrementa la eficiencia de las subidas a los buses pero genera una dificultad para determinar cual bus de todos los que allí se detienen tomó un usuario. Es importante decir que estas estaciones de buses operan durante los horarios de alta demanda en puntos de congestión identificados previamente.

Todas las transacciones bip! Son guardadas en una base de datos que contiene información sobre los operadores y el instante en que la transacción fue hecha. Lo anterior se lleva a cabo por cada pasajero acercando su tarjeta al validador cuando ingresa al bus, zona paga o metro. Cada validador adjunta a cada transacción que realiza un id asociado a el y que está a su vez, asociado con un bus, zona paga o metro. La información recolectada por cada

⁸Un operador es una empresa que se encarga de prestar servicio a una zona de santiago.

transacción incluye: **id de la tarjeta, tipo⁹, código de bus o sitio donde se realizó la transacción, fecha y hora, monto de pago**. La posición espacial de la transacción puede ser conocida directamente para las zonas paga y las estaciones de metro dado que son conocidas con anterioridad, para las transacciones hechas en buses es posible pero no está disponible en la base de datos de transacciones.

Otra base de datos contiene información sobre la localización de todos los buses, como la **latitud, longitud, tiempo, fecha y velocidad instantánea**. Estos datos son obtenidos a intervalos de 30 segundos y son asociados a cada bus a través de un número de placa y código de operador.

Cruzando la información de transacción y posición de las bases de datos por cada placa de bus o código de metro/zona paga y tiempo, es posible identificar la localización espacial donde la transacción es realizada. Es así como en datos analizados del año 2009 y 2010 se logra una estimación en el 98,5 % y 99,9 % de los casos respectivamente.[7].

5.2.2. Estimación de bajadas

Como en el sistema de tarifa solo se validan las subidas, es necesario estimar los puntos de bajadas de las transacciones. Es aquí donde Munizaga y Palma (2012)[7] utilizan una serie de supuestos para entender el comportamiento general de los usuarios dentro del sistema. Para lo anterior utilizan la definición de viaje dada por *Ortúzar y Willumsen, 2011* [9]:

“Un viaje se define como un movimiento desde un punto de origen a un punto de destino donde se realiza una actividad”

Esta definición origina la consideración de etapas dentro de un viaje, una etapa es la utilización de un servicio en particular (bus o metro), entonces la combinación de estos hecha hasta que el usuario llegue al lugar donde debe llevar a cabo su actividad es lo que entendemos como un viaje. Es importante notar que no se consideran los cambios entre líneas de metro.

Básicamente la idea es seguir una cadena de viajes de una tarjeta e identificar la posición de bajada (de bus o metro) mirando la posición y el tiempo de la próxima subida de esta tarjeta. Esto es solamente posible cuando la actual y siguiente transacción tiene información de posición, la cual es tomada de la base de datos de localización automática de vehículos. En el caso de la

⁹puede ser comercial o estudiante.

última transacción del día, se asume que el destino es cercano al punto donde el primer viaje del día comienza, encontrando así un viaje cíclico diario para los usuarios particulares. Si hay solo un viaje por tarjeta, no es posible inferir con solo un día de información.

Los supuestos para llevar a cabo esta estimación son:

- Después de un viaje, el origen del siguiente determina el destino del primero. [2]
- al final del día, los usuarios van a volver a la estación donde abordaron en el primer viaje del mismo día. [2]
- Cada tarjeta corresponde a un usuario. [7]
- Se asume que una persona camina hasta la siguiente parada un máximo de 1.000 metros [7]
- Si el tiempo de transbordo¹⁰ es inferior a 30 minutos, entonces este último servicio forma parte del mismo viaje, de lo contrario se considera como uno nuevo.[7]

Según todo lo anterior Munizaga y Palma (2012)[7] son capaces de estimar cerca del 80 % de los datos utilizados. Dado que es necesario conocer esta información, se restringe los datos al porcentaje de datos que forman parte del resultado exitoso.

5.2.3. Descripción de los datos

El procedimiento anterior ya se encuentra realizado para los datos comprendidos entre el 14 de abril de 2013 al 20 de abril de 2013 por lo que será este tramo el que será utilizado para realizar el análisis espacial y dado que es requerido conocer cada etapa que realiza una persona, se opta por utilizar la tabla de etapas en donde una fila representa una etapa de un viaje determinado. Esta tabla actualmente contiene 42 columnas producto de diversos análisis que se han realizado con ellas pero que no son todos útiles para el desarrollo de esta memoria por lo que se omite la mayoría. En la tabla 1 se listan los campos a ser utilizados para el procesamiento de los datos.

¹⁰entendido como el tiempo entre que se bajó de un servicio y se subió al siguiente.

Nombre campo	Descripción
id	identificador de la tarjeta que realizó la validación
nviaje	Indica el número de viaje asociado al id de la tarjeta de pago
netapa	lugar que ocupa la etapa dentro de un viaje.
par_subida	Indica el paradero en donde abordo el servicio.
par_bajada	Señala el paradero de donde descendió del servicio.

Tabla 1: Atributos a utilizar de la tabla ETAPAS

:

No se descarta que una vez realizada la etapa de visualización puedan requerirse más datos con respecto a cada etapa, como puede ser la fecha y hora, tipo de transporte, entre otros.

Además de la tabla de etapas es necesario conocer todos los paraderos disponibles con el fin de poder realizar la interpolación, por lo que también se utiliza la tabla REDPARADAS (ver tabla 2) en conjunto con la tabla ESTACIONES_METRO (ver tabla 3) que contiene los datos de las estaciones de metro, éstos serán tratados de manera indistinta ya que tanto los paraderos de bus como las estaciones de metro funcionan como puntos de origen o destino en una etapa.

La tabla REDPARADAS contiene 23 columnas con una serie de datos asociados a cada paradero existente en sistema público de transporte de Santiago de Chile. Muchas de éstas han sido agregadas para diversos análisis previos que no se relacionan con esta memoria. Para efectos de este trabajo esta tabla presenta 3 atributos importantes a destacar descritos en la tabla 2.

La tabla ESTACION_METRO contiene 20 atributos, y al igual que la tabla REDPARADAS, la mayoría de sus campos fueron agregados para la ejecución de análisis previos. Los atributos relevantes son descritos en la tabla 3.

Nombre campo	Descripción
codigousuario	identificador único de paradero, ej:PJ156
x	posición en eje horizontal dentro del sistema de coordenadas cartesiana utilizado para la georeferenciación
y	posición en eje vertical dentro del sistema de coordenadas cartesiana utilizado para la georeferenciación

Tabla 2: Atributos a utilizar de la tabla REDPARADAS

:

Nombre campo	Descripción
codigosinlinea	Nombre de la estación de metro
x	posición en eje horizontal dentro del sistema de coordenadas cartesiana utilizado para la georeferenciación
y	posición en eje vertical dentro del sistema de coordenadas cartesiana utilizado para la georeferenciación

Tabla 3: Atributos a utilizar de la tabla ESTACION_METRO

:

5.3. Conclusiones

Es importante destacar que la particularidad del estudio radica en que el grafo formado para construir las estructuras espaciales no representa viajes sino actividades urbanas que realizan las personas, construyendo así una red social formada por actividades urbanas.

Tanto las definiciones de propiedades globales como las formulas definidas aquí serán ocupadas tal cual aparecen de manera de más adelante poder comparar los resultados obtenidos con los mostrados en el artículo, que está basado en el sistema de transporte público de Singapur.

El *factor de amortiguación* será ajustado al propuesto por Chen Zhong et al. (2014)[4], no es campo de esta memoria el considerar variaciones de

este factor, por lo tanto será fijado en 0,85.

Se utilizará el framework *map equation* para llevar a cabo la generación de comunidades dado que plantea el mejor desempeño según Lancichinetti y Fortunato (2009) [6].

La variante de interpolación escogida, al igual que en Chen Zhong et.al (2014) corresponde a la *Inverse Distance Weighting (IDW)* con un $\lambda = 2$ dado que no se hacen supuestos adicionales a los propuestos.

Este trabajo al igual que el artículo en el que está basado, utiliza los datos generados a partir del uso de tarjetas inteligentes del transporte público, con la salvedad de que aquí corresponden al sistema de transporte de Santiago de Chile, con estos se construirá la red espacial para la posterior construcción de las estructuras descritas (centros, centros de flujo y bordes).

Como los datos de Santiago de Chile presentan las mismas características luego de aplicar el método de estimación de Munizaga y Palma (2012), se observa que estos son aptos para replicar el análisis hecho en Singapur. Es importante decir que este procesamiento fue realizado con anterioridad a este trabajo y aquí se documenta con fines explicativos.

La efectividad de la visualización depende de las preferencias y nivel de entrenamiento de los usuarios por lo que se debe considerar la elaboración del perfil de los usuarios que usarán la aplicación.

Al momento de diseñar la visualización no es necesario considerar la distorsión producida en las imágenes bidimensionales presentes en un mapa producto de la proyección existente (3D a 2D) dado que el nivel de detalle que se requiere en las vistas se relaciona a nivel de ciudad, no de calle.

La estrategia para el proceso de visualización queda establecida en la sección 5.1.2 “Construcción y representación de la red”, por lo que no es necesario definir una.

Para la realización de las vistas se debe tener presente la percepción de los usuarios, la intención de ésto y como estos aspectos afecta cada visualización.

En estos momentos existen dos vistas candidatas a ser desplegadas en la versión final. Estas son:

- Visualización del mapa de la ciudad de Santiago de Chile donde se muestra una partición de ésta por el procedimiento que genera las comunidades.
- Visualización del mapa de la ciudad de Santiago de Chile que indica aquellas áreas que son consideradas centros y centros de flujo.

A partir de todo lo expuesto, el aporte de este trabajo a lo ya existente se resume en:

- Validación del modelo propuesto por Cheng Zhong et al. (2014) [4] por medio de la aplicación de su metodología a otra fuente de datos real (sistema de transporte público de Santiago de Chile).
- Diseño de una herramienta de visualización que permite observar los resultados obtenidos por la metodología de Cheng Zhong et al.
- Contribuir al análisis de la información provista por los datos pasivos del transporte público de Santiago de Chile con una herramienta que permite observar una perspectiva diferente a lo realizado previamente.

6. Estrategia de procesamiento

En esta sección se describen las acciones llevadas a cabo para obtener los archivos necesarios para ser analizados y visualizados, es decir, para elaborar el grafo y calcular los indicadores descritos en la sección 5.1. Para lo anterior primero se detallan las herramientas que se utilizarán y luego se explican las acciones llevadas a cabo.

6.1. Herramientas a utilizar

Todos los procesamientos son llevados a cabo en un Notebook con sistema operativo Debian GNU/Linux 7.8 con 3GB de memoria RAM, 40 GB de espacio en disco duro de 5400RPM y un procesador Intel Core i5 4200M de 2.5 GHz. La base de datos que contiene las etapas, paraderos y estaciones de metro se encuentra en un Sistema Gestor de Base de Datos (*SGBD*) **Postgres** versión 9.1 y para gestionar los datos se posee una interfaz gráfica (*GUI*) llamada *PgAdmin III* cuya versión es 1.18.1 con la cuál se manipulan los distintos requerimientos al SGBD. La construcción de comunidades será realizada mediante la herramienta *mapEquation*¹¹.

Por último, se usa el paquete *igraph*¹² como librería del lenguaje C para calcular los indicadores $C_{intermediacion}$ y *PageRank*, además de las propiedades globales del grafo.

Aquellas funcionalidades adicionales que se requieran serán hechas con alguna de las herramientas aquí nombradas o mediante scripts hechos en el lenguaje de programación C o PHP.

La siguiente sección (y en adelante) da por hecho la existencia de una base de datos de nombre 'memoria' que contiene las siguientes tablas:

- etapas: Registros de un tramo realizado
- viajes: Registros de viajes (conjunto de etapas).
- red_paradas: Registros de paraderos.
- etapas_utiles: Registros de la tabla etapa que aprueban el filtrado de datos.

¹¹<http://www.mapequation.org>

¹²<http://igraph.sourceforge.net/>

- viajes_utiles: Registros de la tabla viaje que aprueban el filtrado de datos.
- pajek_por_hora: Registros utilizados para generar los archivos pajek.

Para ver la estructura en detalle de cada tabla revisar anexo 7.0.2.

6.2. Filtrado de datos

En este paso se quitan aquellos datos que no cumplen con el requisito de tener un paradero de bajada (o subida) asociado, para esto se extraen los datos que cumplen con la siguiente consulta en lenguaje *SQL* (*Structure Query Language*) y se guardan en la tabla con datos ya filtrados.

```
1      INSERT INTO etapa_util
2      SELECT id, nviaje, netapa, par_subida, par_bajada
3      FROM etapas
4      WHERE par_subida IS NOT NULL OR par_bajada IS NOT
         NULL;
```

Figura 5: Consulta para quitar etapas sin paradero de bajada.

Los resultados de la consulta anterior (figura 5) son almacenados en la tabla *etapa_util* que contiene solo los campos a ser utilizados.

Se realiza el mismo procedimiento para la tabla de viajes como se puede observar a continuación (figura 6) pero guardandose en una tabla distinta llamada *viaje_util*.

```
1      INSERT INTO viaje_util
2      SELECT codigousuario, x, y
3      FROM redparadas
4      WHERE x IS NOT NULL AND y IS NOT NULL;
```

Figura 6: Consulta para quitar atributos innecesarios de la tabla REDPARADAS.

```
1  SELECT codigousuario , x, y
2  FROM redparadas
3  WHERE x IS NOT NULL AND y IS NOT NULL;
```

Figura 7: Consulta para quitar atributos innecesarios de la tabla REDPARADAS.

```
1  SELECT codigosinlinea , x, y
2  FROM estacion_metro
3  WHERE x IS NOT NULL AND y IS NOT NULL;
```

Figura 8: Consulta para quitar atributos innecesarios de la tabla ESTACION_METRO.

Al finalizar esta etapa se obtienen 3 tablas con los datos necesarios para llevar a cabo el cálculo de las siguientes instrucciones.

6.3. Construcción de la red

Con los datos de la tabla ETAPA se construye una matriz origen destino donde cada celda C_{ij} denota el volumen de etapas realizadas desde el área asociada al paradero i al área asociada al paradero j . Lo anterior se realizará utilizando la herramienta R con el paquete *i-graph*.

6.4. Trabajo pendiente

A continuacion se listan las tareas pendientes de esta sección:

- Cálculo de centralidades.
 - Centralidad de intermediación
 - PageRank
- Cálculo de bordes y generación de comunidades
- Realización de interpolación espacial
- Corrección de bordes

Como se dijo anteriormente, serán realizadas durante el período previo al inicio del semestre 2015-1.

6.5. Solo punteos

Antes de hacer cualquier cosa con los datos debemos eliminar aquellos que no tienen paradero de subida ni de bajada.

Listing 1: Generación de arcos

```
1 DELETE FROM etapas WHERE par_subida IS NULL OR
   par_bajada IS NULL;
```

La eliminación demoró 879228.976 ms (14.65 m) y sequitaron 6.963.870 registros sin paradero de bajada y 5434 que no tienen paradero de subida.

consulta para obtener los arcos con pesos

Listing 2: Generación de arcos

```
1 su postgres
2
3 psql memoria
4
5 — lo guardamos como un CSV
6 copy (SELECT par_subida , par_bajada , COUNT(*) AS peso
7        FROM etapas AS prueba
8        GROUP BY par_subida , par_bajada)
9 To '/var/tmp/dia.csv' with CSV;
10
11
12 — V2 lo guardamos como un CSV. No funciona parece que
   hay codigos que no estan en la tabla de paraderos.
   Esta consulta no se ocupo (con acento en la o)
13 copy (SELECT subida.nombre, bajada.nombre, arcos.peso
14        FROM (SELECT par_subida , par_bajada , COUNT(*) AS
15               peso
16               FROM etapas AS prueba
17               GROUP BY par_subida , par_bajada) AS arcos
18        LEFT JOIN redparadas AS subida ON arcos.
19        par_subida = subida.codigo
```

```

18      LEFT JOIN redparadas AS bajada ON arcos.
        par_subida = bajada.codigo)
19 To '/var/tmp/dia.csv' with CSV;

```

La ejecución de esta consulta demoró 2187691.139 ms (36.46 m) y se generaron 644.761 filas, lo que indica la cantidad de arcos presentes. Esto es sobre todos los viajes existentes.

Ahora se genera el mismo archivo pero con un rango horario de mañana (05:00-12:00) y otro rango horario de tarde (16:00-21:00). Para lo anterior se generan dos índices para mejorar el rendimiento de la consulta.

Antes de consultar se crea un índice sobre el campo tiempo_subida para mejorar el rendimiento.

Listing 3: Creación de índice

```

1 CREATE INDEX hora_subida ON etapas (extract(hour from
    tiempo_subida));

```

Su creación demoró 294591.532 ms (4.91 min)

Luego se genera el archivo con datos de la mañana y tarde según los períodos definidos mas arriba.

Listing 4: Generación de arcos horario mañana

```

1 su postgres
2
3 psql memoria
4
5 // lo guardamos como un CSV
6 copy (SELECT par_subida, par_bajada, COUNT(*) AS peso
7      FROM etapas AS prueba
8      WHERE extract(hour from tiempo_subida) BETWEEN 05
9            AND 12
10     GROUP BY par_subida, par_bajada)
10 To '/var/tmp/manana.csv' with CSV;

```

demoró 1221570.569 ms (20.34 min) y generó 457.015 arcos.

Listing 5: Generación de arcos horario tarde

```

1 su postgres
2
3 psql memoria

```

```

4
5 // lo guardamos como un CSV
6 copy (SELECT par_subida, par_bajada, COUNT(*) AS peso
7       FROM etapas AS prueba
8       WHERE extract(hour from tiempo_subida) BETWEEN 16
9           AND 21
9       GROUP BY par_subida, par_bajada)
10 To '/var/tmp/tarde.csv' with CSV;

```

demoró 953715.972 ms (15.9 min) y generó 438.672 arcos.

Formato de archivo

Ahora debemos crear el archivo de nodos, para ellos ocuparemos el formato PAJEK y uno script (crear_archivo_pajek.php) en php convierten el CSV creado en un archivo con ese formato.

Investigar—

Ahora debemos intercambiar los nombres por números. Para lo anterior ocuparemos la tabla redparadas. Vamos a correr el script `cambiar_nombre_pajek.php`.

correr infomap (por el momento estamos ocupando la app web)

`./infomap -input-format=" -d -2`

opciones `-ijsi`, `-input-format=jsi`: Specify input format ('pajek', 'link-list', '3gram' or 'multiplex') to override format possibly implied by file extension. `-map`: Print the top two-level modular network in the .map format. `-clu`: Print the top cluster indices for each node. `-2`, `-two-level`: Optimize a two-level partition of the network. `-d`, `-directed`: Assume directed links.

el calculo de lo indicadores se hace con el programa Gephi.

Explicar como conseguirlos y exportar los datos en un csv.

7. Bibliografía

- [1] Alex Anas, Richard Arnott, Kenneth A. Small. Urban spatial structure. *Journal of Economic Literature*, 36:1426–1464, 1998.
- [2] Barry, J.J., Newhouser, R. Rahbee, A., Sayeda, S. Origin and destination estimation in new york city with automated fare system data. *Transportation research record*, 1817:183–187, 2002.
- [3] Cardillo, A., et al. Structural properties of planar graphs of urban street patterns. *Physical Review*, 73, 2006.
- [4] Chen Zhong, Stefan Müller Arisona, Xianfeng Huang, Michael Batty & Gerhard Schmitt. Detecting the dynamics of urban structure through spatial network analysis. *International Journal of Geographical Information Science*, 50:1–21, 2014.
- [5] Doursat, R. Topology and dynamics of complex networks. In: *CS 790R Seminar modeling & simulation. Department of Computer Science & Engineering University of Nevada, Reno, Spring*, 2005.
- [6] Lancichinetti, A. and Fortunato, S. Community detection algorithms: a comparative analysis. *Physical review E*, 2009.
- [7] Marcela A. Munizaga, Carolina Palma. Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from santiago, chile. *Transportation Research Part C: Emerging Technologies*, 24:9–18, 2012.
- [8] Matthew O. Ward, Georges Grinstein, Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, 2010.
- [9] Ortuzar, J. de D., Willumsen, L.G. *Modelling transport*. Wiley, Chichester, 2011.
- [10] Ratti, C., et al. Redrawing the map of great britain from a network of human interactions. *PloS One*, 5, 2010.
- [11] Rinzivillo, S., et al. Discovering the geographical borders of human mobility. *KI-Künstliche Intelligenz*, 26:253–260, 2012.

- [12] Rosvall, M. y Bergstrom, C.T. Maps of random walks on complex networks. *PloS One*, 5, 2008.
- [13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.
- [14] Soh, H., et al. Weighted complex network analysis of travel routes on the singapore public transportation system. *Physica A: Statistical Mechanics and its Applications*, 389:5852–5863, 2010.

7.0.1. Instalación de Postgres

Para instalar postgres en Debian GNU/Linux se deben ejecutar la siguiente secuencia de operaciones en la consola de comandos.

Listing 6: Instalacion de postgres

```
1 # El usuario con el que se ejecuta este codigo
2 # debe ser root.
3
4 # Instala el sistema gestor de base de datos Postgres,
5 # un programa para interactuar con las bases de datos
6 # (psql) y crea un usuario en el sistema de nombre
7 # postgres.
8 apt-get install postgresql postgresql-client
```

7.0.2. Creación de base de datos

El código mostrado a continuación crea la estructura de la base de datos requerida para este trabajo.

Listing 7: Instalacion de postgres

```
1 # Se inicia sesion con el usuario postgres
2 su postgres
3 # Se accede al programa para interactuar con la base de
  datos.
4 psql
5
6 # Se crea usuario y base de datos
7 CREATE USER felipe WITH PASSWORD 'felipe';
8 CREATE DATABASE memoria OWNER felipe;
9
10 # Se cierra psql
11 \q
12
13 # Se ingresa a la base de datos 'memoria'
14 psql memoria
15
16 — crear tabla etapas
17 CREATE TABLE etapas (
```


18 tiempo_subida **timestamp** without **time zone**,
19 id **bigint**,
20 pago **smallint**,
21 x_subida double precision,
22 y_subida double precision,
23 tipo_transporte **character varying**(20),
24 serviciosentidovariante **character varying**(500),
25 tipo_dia **character varying**(10),
26 nviaje **integer**,
27 netapa **integer**,
28 x_bajada double precision,
29 y_bajada double precision,
30 tiempo_bajada **timestamp** without **time zone**,
31 par_subida **character varying**(30),
32 par_bajada **character varying**(30),
33 comuna_subida **character varying**(20),
34 comuna_bajada **character varying**(20),
35 zona_subida **character varying**(10),
36 zona_bajada **character varying**(10),
37 distonroutesubidabajada **integer**,
38 disteuclidsubidabajada **integer**,
39 distonrouteparsubidaparabajada **integer**,
40 disteuclidparsubidaparabajada **integer**,
41 serv_un_zp2 **character varying**(500),
42 sitio2 **character varying**(50),
43 tiempo2 **timestamp** without **time zone**,
44 media_hora **time** without **time zone**,
45 tiempo_trasbordo double precision,
46 dist_trasbordo double precision,
47 tiempo_caminata double precision,
48 tiempo_espera double precision,
49 tiempo_etapa double precision,
50 tiempo_espera_estimado double precision,
51 adulto **integer**,
52 factor_exp_etapa double precision,
53 nbusesanteriores **integer**,
54 tiempo_busanterior **timestamp** without **time zone**,
55 busanterior1er **integer**,

```
56     busanterior2do integer ,
57     busanterior3er integer ,
58     tipocorte character varying(100) ,
59     proposito character varying(100)
60 );
61
62 ALTER TABLE public.etapas OWNER TO felipe;
63
64 — crear tabla de viajes
65 CREATE TABLE viajes (
66     id character varying(20) ,
67     nviaje integer ,
68     netapa integer ,
69     etapas character varying(1000) ,
70     netapassinbajada integer ,
71     ultimaetapaconbajada integer ,
72     tviaje_seg double precision ,
73     tviaje_min double precision ,
74     dviajeeuclidiana_mts double precision ,
75     dviajeenruta_mts double precision ,
76     paraderosubida character varying(50) ,
77     paraderobajada character varying(50) ,
78     comunasubida character varying(20) ,
79     comunabajada character varying(20) ,
80     diseno777subida character varying(20) ,
81     diseno777bajada character varying(20) ,
82     tiemposubida timestamp without time zone ,
83     tiempobajada timestamp without time zone ,
84     periodosubida character varying(100) ,
85     periodobajada character varying(100) ,
86     tipodia character varying(10) ,
87     mediahora time without time zone ,
88     contrato character varying(50) ,
89     factorexpancion double precision ,
90     tiempomediodeviaje timestamp without time zone ,
91     periodomediodeviaje character varying(200) ,
92     mediahoramediodeviaje time without time zone ,
93     tipodiamediodeviaje character varying(10) ,
```

94 t_1era_etapa double precision ,
95 d_1era_etapa double precision ,
96 tespera_1era_etapa double precision ,
97 ttrasbordo_1era_etapa double precision ,
98 tcaminata_1era_etapa double precision ,
99 t_2da_etapa double precision ,
100 d_2da_etapa double precision ,
101 tespera_2da_etapa double precision ,
102 ttrasbordo_2da_etapa double precision ,
103 tcaminata_2da_etapa double precision ,
104 t_3era_etapa double precision ,
105 d_3era_etapa double precision ,
106 tespera_3era_etapa double precision ,
107 ttrasbordo_3era_etapa double precision ,
108 tcaminata_3era_etapa double precision ,
109 t_4ta_etapa double precision ,
110 d_4ta_etapa double precision ,
111 tespera_4ta_etapa double precision ,
112 ttrasbordo_4ta_etapa double precision ,
113 tcaminata_4ta_etapa double precision ,
114 op_1era_etapa **character varying(10)** ,
115 op_2da_etapa **character varying(10)** ,
116 op_3era_etapa **character varying(10)** ,
117 op_4ta_etapa **character varying(10)** ,
118 tipoop_1era_etapa **character varying(15)** ,
119 tipoop_2da_etapa **character varying(15)** ,
120 tipoop_3era_etapa **character varying(15)** ,
121 tipoop_4ta_etapa **character varying(15)** ,
122 serv_1era_etapa **character varying(500)** ,
123 serv_2da_etapa **character varying(500)** ,
124 serv_3era_etapa **character varying(500)** ,
125 serv_4ta_etapa **character varying(500)** ,
126 linea_metro_subida_1 **character varying(100)** ,
127 linea_metro_subida_2 **character varying(100)** ,
128 linea_metro_subida_3 **character varying(100)** ,
129 linea_metro_subida_4 **character varying(100)** ,
130 linea_metro_bajada_1 **character varying(100)** ,
131 linea_metro_bajada_2 **character varying(100)** ,

132 linea_metro_bajada_3 **character varying**(100),
133 linea_metro_bajada_4 **character varying**(100),
134 paraderosubida_1era **character varying**(50),
135 paraderosubida_2da **character varying**(50),
136 paraderosubida_3era **character varying**(50),
137 paraderosubida_4ta **character varying**(50),
138 tiemposubida_1era **timestamp** without **time zone**,
139 tiemposubida_2da **timestamp** without **time zone**,
140 tiemposubida_3era **timestamp** without **time zone**,
141 tiemposubida_4ta **timestamp** without **time zone**,
142 zona777subida_1era **character varying**(50),
143 zona777subida_2da **character varying**(50),
144 zona777subida_3era **character varying**(50),
145 zona777subida_4ta **character varying**(50),
146 paraderobajada_1era **character varying**(50),
147 paraderobajada_2da **character varying**(50),
148 paraderobajada_3era **character varying**(50),
149 paraderobajada_4ta **character varying**(50),
150 tiempobajada_1era **timestamp** without **time zone**,
151 tiempobajada_2da **timestamp** without **time zone**,
152 tiempobajada_3era **timestamp** without **time zone**,
153 tiempobajada_4ta **timestamp** without **time zone**,
154 zona777bajada_1era **character varying**(50),
155 zona777bajada_2da **character varying**(50),
156 zona777bajada_3era **character varying**(50),
157 zona777bajada_4ta **character varying**(50),
158 tipotransporte_1era **character varying**(50),
159 tipotransporte_2da **character varying**(50),
160 tipotransporte_3era **character varying**(50),
161 tipotransporte_4ta **character varying**(50),
162 tespera_estimada_1era **double precision**,
163 tespera_estimada_2da **double precision**,
164 tespera_estimada_3era **double precision**,
165 tespera_estimada_4ta **double precision**,
166 escolar **character varying**(10),
167 tviaje_en_vehiculo_min **double precision**,
168 tipo_corte_etapa_viaje **character varying**(50),
169 proposito **character varying**(50),

```
170     dviaje_buses double precision
171 );
172
173 ALTER TABLE public.viajes OWNER TO felipe;
174
175 — crear tabla de paraderos (no incluye estaciones de
    metro)
176 CREATE TABLE redparadas (
177     codigo character varying(20),
178     codigousuario character varying(20),
179     comuna character varying(100),
180     nombre character varying(500),
181     sentido character varying(500),
182     fila_superior character varying(500),
183     fila_inferior character varying(500) —,
184     grupo_parada character varying(500),
185     x integer,
186     y integer,
187     latitud double precision,
188     longitud double precision,
189     censal_1992 character varying(20),
190     comunas character varying(20),
191     diseno_563 character varying(20),
192     diseno_777 character varying(20),
193     eod_2001 character varying(20),
194     eod_2006 character varying(20),
195     estraus_264 character varying(20),
196     estraus_404 character varying(20),
197     estraus_410 character varying(20),
198     estraus_618 character varying(20),
199     zonas_6 character varying(20)
200 );
201
202 ALTER TABLE public.redparadas OWNER TO felipe;
203
204 — crear tabla etapa_util
205 CREATE TABLE etapa_util (
206     id bigint,
```

```
207     nviaje integer ,
208     netapa integer ,
209     par_subida character varying(30) ,
210     par_bajada character varying(30)
211 );
212
213 ALTER TABLE public.etapa_util OWNER TO felipe ;
214
215 — crear tabla de viaje_util
216 CREATE TABLE viaje_util (
217     id character varying(20) ,
218     nviaje integer ,
219     netapa integer ,
220     etapas character varying(1000) ,
221     netapassinbajada integer ,
222     ultimaetapaconbajada integer ,
223     paraderosubida character varying(50) ,
224     paraderobajada character varying(50) ,
225     tiemposubida timestamp without time zone ,
226     tiempobajada timestamp without time zone ,
227     tipodia character varying(10) ,
228     mediahora time without time zone ,
229     contrato character varying(50) ,
230     factorexpancion double precision ,
231     tiempomediodeviaje timestamp without time zone ,
232     periodomediodeviaje character varying(200) ,
233     mediahoramediodeviaje time without time zone ,
234     tipodiamediodeviaje character varying(10) ,
235     t_1era-etapa double precision ,
236     d_1era-etapa double precision ,
237     tespera_1era-etapa double precision ,
238     ttrasbordo_1era-etapa double precision ,
239     tcaminata_1era-etapa double precision ,
240     t_2da-etapa double precision ,
241     d_2da-etapa double precision ,
242     tespera_2da-etapa double precision ,
243     ttrasbordo_2da-etapa double precision ,
244     tcaminata_2da-etapa double precision ,
```

245 t_3era_etapa double precision ,
246 d_3era_etapa double precision ,
247 tespera_3era_etapa double precision ,
248 ttrasbordo_3era_etapa double precision ,
249 tcaminata_3era_etapa double precision ,
250 t_4ta_etapa double precision ,
251 d_4ta_etapa double precision ,
252 tespera_4ta_etapa double precision ,
253 ttrasbordo_4ta_etapa double precision ,
254 tcaminata_4ta_etapa double precision ,
255 op_1era_etapa **character varying**(10) ,
256 op_2da_etapa **character varying**(10) ,
257 op_3era_etapa **character varying**(10) ,
258 op_4ta_etapa **character varying**(10) ,
259 tipoop_1era_etapa **character varying**(15) ,
260 tipoop_2da_etapa **character varying**(15) ,
261 tipoop_3era_etapa **character varying**(15) ,
262 tipoop_4ta_etapa **character varying**(15) ,
263 serv_1era_etapa **character varying**(500) ,
264 serv_2da_etapa **character varying**(500) ,
265 serv_3era_etapa **character varying**(500) ,
266 serv_4ta_etapa **character varying**(500) ,
267 linea_metro_subida_1 **character varying**(100) ,
268 linea_metro_subida_2 **character varying**(100) ,
269 linea_metro_subida_3 **character varying**(100) ,
270 linea_metro_subida_4 **character varying**(100) ,
271 linea_metro_bajada_1 **character varying**(100) ,
272 linea_metro_bajada_2 **character varying**(100) ,
273 linea_metro_bajada_3 **character varying**(100) ,
274 linea_metro_bajada_4 **character varying**(100) ,
275 paraderosubida_1era **character varying**(50) ,
276 paraderosubida_2da **character varying**(50) ,
277 paraderosubida_3era **character varying**(50) ,
278 paraderosubida_4ta **character varying**(50) ,
279 tiemposubida_1era **timestamp** without **time zone** ,
280 tiemposubida_2da **timestamp** without **time zone** ,
281 tiemposubida_3era **timestamp** without **time zone** ,
282 tiemposubida_4ta **timestamp** without **time zone** ,

```
283     zona777subida_1era character varying(50) ,
284     zona777subida_2da character varying(50) ,
285     zona777subida_3era character varying(50) ,
286     zona777subida_4ta character varying(50) ,
287     paraderobajada_1era character varying(50) ,
288     paraderobajada_2da character varying(50) ,
289     paraderobajada_3era character varying(50) ,
290     paraderobajada_4ta character varying(50) ,
291     tiempobajada_1era timestamp without time zone ,
292     tiempobajada_2da timestamp without time zone ,
293     tiempobajada_3era timestamp without time zone ,
294     tiempobajada_4ta timestamp without time zone ,
295     zona777bajada_1era character varying(50) ,
296     zona777bajada_2da character varying(50) ,
297     zona777bajada_3era character varying(50) ,
298     zona777bajada_4ta character varying(50) ,
299     tipotransporte_1era character varying(50) ,
300     tipotransporte_2da character varying(50) ,
301     tipotransporte_3era character varying(50) ,
302     tipotransporte_4ta character varying(50) ,
303     tespera_estimada_1era double precision ,
304     tespera_estimada_2da double precision ,
305     tespera_estimada_3era double precision ,
306     tespera_estimada_4ta double precision ,
307     escolar character varying(10) ,
308     tviaje_en_vehiculo_min double precision ,
309     tipo_corte_etapa_viaje character varying(50) ,
310     proposito character varying(50) ,
311     dviaje_buses double precision
312 );
313
314 ALTER TABLE public.viaje_util OWNER TO felipe ;
315
316 — crear tabla pajek_por_hora
317 CREATE TABLE pajek_por_hora (
318     id bigint ,
319     nviaje integer ,
320     netapa integer ,
```



```
321     par_subida character varying(30) ,
322     par_bajada character varying(30) ,
323     hora character varying(5) — se indica la hora de
      subida 1,2,...,23 o un rango 1-2, 21-00, ...
324 );
```

7.0.3. Importación de los datos

Los datos fueron extraídos de la base de datos del departamento de Ingeniería Civil Transporte de la FCFM mediante el comando *pg_dump* de postgres y almacenados en un archivo de texto plano. A continuación se nombran los archivos que contienen los datos:

backup_etapas (12.1 GB): Contiene los datos de todas las etapas realizadas entre 14 y 20 de abril de 2013.

backup_viajes (16 GB): Contiene los datos de todas los viajes realizados entre 14 y 20 de abril de 2013.

backup_redparadas (1.2 MB): Contiene los datos de todos los paraderos presentes en Transantiago.

Listing 8: Importacion de datos

```
1 # Se inicia sesion con el usuario postgres
2 su postgres
3
4 # Se cargan los datos en la base de datos
5 psql -d memoria -a -f /ruta/backup_etapas
6 psql -d memoria -a -f /ruta/backup_viajes
7 psql -d memoria -a -f /ruta/backup_redparadas
8
9 # se ingresa a la instancia de la base de datos
10 psql memoria
```

7.0.4. Instalación Infomap

Infomap es el framework que reliaza la tarea de detectar las comunidades. Para acceder a el se debe descargar el archivo .zip del siguiente link

<https://bitbucket.org/mapequation/infomap/get/v0.17.2.zip>
luego ejecutamos la siguiente secuencia de comandos

Listing 9: Instalación infomap

```
1 # nos posicionamos en la carpeta de descarga
2 cd /ruta/del/archivo/descargado
3
4 # descomprimir archivo
5 unzip Infomap.zip
6
7 # ingresar a la carpeta obtenida en la descompresion
8 cd mapequation-infomap-d462e7f4a0f7
9
10 # crear el ejecutable
11 make
```

7.0.5. Instalación de R

Para instalar R en Debian GNU/Linux se deben ejecutar la siguiente secuencia de comandos en la consola de comandos.

Listing 10: Instalacion de R

```
1 # Instala los archivos base
2 apt-get install r-base
3
4 # Se ingresa al entorno de R
5 R
6
7 # Se instala el paquete i-graph
8 install.packages("igraph")
```