

Departamento de Investigación Operativa
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo – Uruguay

Proyecto de Grado
Ingeniería en Computación

Interfaz para herramienta de
planificación de recorridos para
transporte público.

Informe final.

Diego Gawenda, Héctor Manuel Martínez

Marzo, 2009.

Tutores:

MSc. Antonio Mauttone

MSc. María Urquhart

Resumen

En la planificación del transporte público se identifica como un problema a resolver el de encontrar un conjunto de recorridos de ómnibus y sus frecuencias asociadas. Para ello existen algoritmos de optimización combinatoria, los cuales requieren ciertos datos de entrada como ser: la red de calles por donde transitan los ómnibus, la demanda de viajes existente, las principales características de la flota de ómnibus y los parámetros del modelo de optimización a ejecutar. En este marco el proyecto consiste en construir una herramienta capaz de representar y manipular gráficamente los recorridos. Para esto la misma debe interactuar con los algoritmos de optimización y con aquella información necesaria para poder ejecutarlos con el fin de comunicar los resultados a los usuarios planificadores. Una parte importante de esa información, es de tipo geográfica. Para su manejo se definió como uno de los requisitos no funcionales que la tecnología de información geográfica a utilizar fuese no propietaria, siendo MapWinGIS la opción seleccionada.

La herramienta se divide en tres módulos independientes: 1) construcción de un caso de estudio, 2) manipulación del caso de estudio y 3) alta y baja de los algoritmos. La construcción de un caso de estudio se divide a su vez en tres etapas: zonificación de la ciudad, construcción de la conectividad y cálculo de la matriz de demanda.

La manipulación del caso de estudio es el proceso de construcción y evaluación de las soluciones; una solución es un conjunto de recorridos y sus frecuencias asociadas.

En este tipo de herramientas es de gran relevancia la interacción con el usuario, por lo tanto se tuvo especial cuidado para que fuese amigable, atractiva y fácil de usar. En este trabajo se utiliza como caso de estudio el sistema de transporte colectivo de la ciudad de Rivera.

Palabras claves: interfaz gráfica, sistemas de información geográfica, planificación de recorridos para el transporte público.

Tabla de Contenidos.

0.	Acerca del documento	6 -
1.	Introducción.....	7 -
1.1	Contexto del proyecto.....	7 -
1.2	El problema del transporte público.....	8 -
1.3	Limitaciones de los proyectos anteriores.....	9 -
1.4	Objetivos del proyecto.....	10 -
1.4.1	Generales.	10 -
1.4.2	Específicos.....	11 -
2.	Desarrollo	12 -
2.1	Requisitos Funcionales	12 -
2.1.1	Módulos	14 -
2.1.2	Algoritmos	18 -
2.2	Requisitos no Funcionales	21 -
2.3	Diseño y Arquitectura.....	22 -
2.4	Interacción con el Usuario	25 -
2.4.1	Organizar.	28 -
2.4.2	Economizar.....	31 -
2.4.3	Comunicar.....	33 -
2.5	Producto final.....	37 -
3.	Conclusiones.....	41 -
3.1	Trabajos a Futuro	42 -
4.	Agradecimientos	45 -
	Bibliografía	46 -
5.	Anexo. Caso de estudio: Ciudad de Rivera	53 -
5.1	Objetivo 4 - Obtención de datos de calidad.....	53 -
5.1.1	Procesamiento de la encuesta de hogares	53 -
5.1.2	Digitalización de capas geográficas de la ciudad	54 -
5.2	Objetivo 6 – Construcción de una propuesta alternativa	56 -
5.2.1	Armado del caso de estudio	56 -
5.2.2	Experimentos con el caso de estudio	63 -
6.	Anexo. Búsqueda y Selección de Tecnologías SIG.....	69 -

6.1	Introducción	- 69 -
6.2	SIG-T	- 69 -
6.2.1	Funciones de análisis de los SIG-T.....	- 70 -
6.3	SIG Libre	- 77 -
6.4	Software libre.....	- 78 -
6.4.1	Movimientos del software libre	- 78 -
6.4.2	Licencias del software libre	- 79 -
6.5	Software Analizado	- 81 -
6.5.1	Herramientas.....	- 81 -
6.5.2	Aplicaciones de escritorio.....	- 82 -
6.5.3	Bibliotecas	- 97 -
6.5.4	Evaluación del Software analizado.....	- 101 -
7.	Anexo. Entorno de Desarrollo	- 102 -
8.	Anexo. Análisis y Diseño	- 106 -
8.1	Modelo de Dominio	- 106 -
8.1.1	Modelo de Dominio - Construcción del Caso de Estudio	- 106 -
8.1.2	Modelo de Dominio - Manipulación del Caso de Estudio.....	- 107 -
8.2	Análisis de requisitos	- 108 -
8.2.1	Resumen	- 108 -
8.2.2	Descripción	- 108 -
8.2.3	Usuarios	- 111 -
8.2.4	Requisitos Funcionales	- 111 -
8.2.5	Requisitos no Funcionales	- 117 -
8.2.6	Interfaces.....	- 117 -
8.3	Casos de Uso.....	- 118 -
8.3.1	Usuario del Sistema	- 118 -
	Diagramas de Caso de Uso	- 119 -
8.3.2	Descripción de los Casos de Uso	- 121 -
8.4	Arquitectura de la Herramienta.....	- 142 -
8.4.1	Funcionalidades de las clases.	- 144 -
9.	Anexo. Verificación y Validación	- 146 -
9.1	Módulo Central	- 146 -

9.2	Módulo Configuración de Algoritmos.....	- 146 -
9.3	Módulo Construcción del Caso de Estudio.....	- 147 -
9.4	Módulo Manipulación del Caso de Estudio.....	- 149 -
10.	Anexo. Prototipos	- 156 -
10.1	Menús	- 157 -
10.2	Escenario de Interacción.....	- 158 -
11.	Anexo. Actas.....	- 166 -
12.	Anexo. Transporte Público	- 175 -
12.1	Planeamiento Estratégico.....	- 175 -
12.2	Diseño de la red	- 175 -
12.2.1	Asignación de pasajeros	- 177 -
12.3	Planeamiento Táctico.....	- 178 -
12.3.1	Ajuste de frecuencias	- 178 -
12.3.2	Construcción de la tabla de horarios.....	- 178 -
12.4	Planeamiento operacional.....	- 180 -
12.4.1	Programación de vehículos.....	- 180 -
12.4.2	Programación de tareas.....	- 182 -
12.4.3	Programación integrada de vehículos y tareas.....	- 183 -
12.4.4	Ordenamiento de tripulación	- 184 -
12.4.5	Estacionamiento y despacho.....	- 184 -
12.4.6	Programación de mantenimiento	- 185 -
12.5	Controles en tiempo real.....	- 186 -
13.	Glosario.....	- 188 -

0. Acerca del documento

Este documento se divide en tres secciones principales y ocho anexos complementarios. En los capítulos iniciales se introduce al lector al problema enfrentado, el cual motiva a este Proyecto de Grado; luego se detallan los requisitos del mismo junto al diseño de la solución. Posteriormente se define la interacción entre el producto construido y sus usuarios. Por último se concluye mostrando los logros conseguidos y aquellos desafíos que quedan planteados para futuros desarrollos.

Los anexos complementan el informe desde distintas perspectivas, ellos son:

- Anexo. Caso de estudio: Ciudad de Rivera. Se describe un escenario típico del uso de la herramienta orientado a que el lector tenga una noción de las características de la misma.
- Anexo. Búsqueda y Selección de Tecnologías SIG. Tiene principalmente un propósito de documentación para futuros trabajos.
- Anexo. Entorno de Desarrollo. Motivación de la selección del entorno de desarrollo.
- Anexo. Análisis y Diseño. Es fundamental para entender como está construida la herramienta.
- Anexo. Verificación y Validación. Casos de prueba experimentados, modificaciones y problemas encontrados por parte del usuario.
- Anexo. Prototipos. Primeros modelos de interacción entre la herramienta y el usuario.
- Anexo. Actas. Documentación acerca de las reuniones efectuadas con el cliente.
- Anexo. Transporte Público. Material complementario acerca de la problemática que involucra a este proyecto.

Otra documentación que existe es el manual de usuario accesible a través de la opción de “Ayuda” de la herramienta de software construida, donde se describen las funcionalidades y la forma de uso de las mismas. También se cuenta con documentación técnica que simplifica la tarea de futuros emprendimientos sobre la herramienta; este texto se encuentra en el CD entregado.

1. Introducción

1.1 Contexto del proyecto

Este proyecto se enmarca en la línea de trabajo del Departamento de Investigación Operativa acerca de la aplicación de metodologías y herramientas de la Investigación Operativa a problemas de transporte. En particular actualmente se desarrolla un proyecto de investigación PDT 48/02 (de ahora en más PDT) y un Doctorado en Informática PEDECIBA en temas de optimización de recorridos y frecuencias en transporte público. El proyecto PDT tiene como objetivos:

1-Desarrollo y aplicación de una metaheurística para el TNDP en un contexto multiobjetivo.

2-Implementación de un modelo de asignación que refleje de manera adecuada, el comportamiento de los usuarios y los costos de los trazados de los recorridos.

3-Desarrollo de una interfaz de usuario, que manipule datos geográficos y de demanda, los datos de demanda en la forma de matrices origen-destino.

4-Obtención de datos de calidad acerca de los viajes realizados utilizando transporte público urbano colectivo en la ciudad de Rivera.

5-Desarrollo de una herramienta de apoyo a la solución del TNDP para el caso de Rivera.

6-Propuesta de un conjunto de soluciones alternativas para el diseño topológico de los recorridos y valores de las frecuencias para el sistema de transporte público urbano colectivo de la ciudad de Rivera. Se tratará de encontrar soluciones con diferentes niveles de compromiso entre los costos para usuarios y operadores, de modo que el planificador pueda elegir la que más le convenga.

7-Estudio de la posibilidad de extender los resultados de este proyecto a ciudades de mayor porte (por ejemplo Montevideo).

El Proyecto de Grado se enmarca en la línea de trabajo del PDT, y la herramienta que se desarrolló es consecuencia de los objetivos 3 y 5.

1.2 El problema del transporte público

Durante años la Investigación Operativa (IO) ha resuelto con éxito una gran variedad de problemas de optimización relacionados con el transporte público. Se han diseñado varios paquetes de software comerciales que se basan en técnicas de la IO, que son utilizados por las agencias de transporte público para ayudar a planificar sus operaciones. Los problemas del transporte público son un gran atractivo para la IO debido a su tamaño, complejidad e interés social. Por ejemplo, el sistema de transporte de la ciudad de Montevideo emplea a más de 5000 empleados para operar aproximadamente 1300 buses que usan 150 recorridos [CU 2008], [CO 2008], [COE 2008], [UC 2008]. Más aún, estos problemas son complejos debido a que involucran pasajeros, buses, y tripulación, los cuales están sujetos a preferencias y limitaciones individuales, e interactúan entre ellos acorde a un conjunto de relaciones preestablecidas.

El objetivo principal de la mayoría de las agencias de transporte público es ofrecer un servicio de calidad a la población, que permita a los pasajeros viajar fácilmente a bajo costo. Además, las agencias tienen la misión social de reducir tanto la contaminación como la congestión de tráfico, así como de incrementar la movilidad de la población. En la mayoría de los casos, el objetivo no es usualmente obtener ganancias, como sí es el caso de otras organizaciones de transporte como las compañías aéreas, ferroviarias y de camiones. Estas están sujetas a restricciones presupuestarias que las fuerzan a manejar recursos costosos como buses, conductores, mantener instalaciones y depósitos lo más eficientemente posible. En resumen, el problema global que enfrentan las agencias consiste en determinar como ofrecer una buena calidad de servicio a los pasajeros manteniendo valores y costos de operación razonables.

Encarado como un todo, el problema global no es tratable. Por lo tanto, se lo dividido en un conjunto de subproblemas que usualmente se resuelven de manera secuencial en varios pasos dentro del proceso en contextos tanto de planificación (estratégico, táctico y operacional), como en operación (control en tiempo real). Los problemas estratégicos se encargan de las decisiones a largo plazo como es el diseño de los recorridos y la red. La mayoría de estos problemas entra en la categoría de diseño de redes y requiere resolver el problema de la asignación de pasajeros como un subproblema o con propósito de evaluación. El objetivo de los problemas estratégicos

es maximizar la calidad del servicio bajo restricciones presupuestarias. Los problemas tácticos se encargan de las decisiones relacionadas con el servicio ofrecido al público, como la frecuencia de los servicios a lo largo de los recorridos y las tablas de horarios de viajes. Estos problemas son usualmente resueltos por temporada, con actualizaciones ocasionales y también se enfocan en la calidad del servicio. Los problemas operacionales se refieren a como las operaciones deben llevarse a cabo para ofrecer el servicio propuesto a un mínimo costo. Incluyen una gran variedad de problemas como programación de vehículos, programación de tripulación, estacionamiento y despacho de buses en los garajes, y programación de mantenimiento. Los problemas operacionales son resueltos en varios intervalos de tiempo en un rango que va desde una vez por mes para la programación de la tripulación a una vez por día para el estacionamiento y despacho de ómnibus en los garajes. En contraste a los objetivos de los problemas previos, el objetivo para los problemas operacionales es claramente el de minimizar el costo total. Por último, los problemas de control en tiempo real manejan las perturbaciones que sufre el plan utilizando varias estrategias de control. Estos problemas son resueltos en tiempo real durante las operaciones con el objetivo de minimizar los inconvenientes causados a los pasajeros. Usualmente, son mínimos o no existen los costos de operación que involucran ya que estos consideran perturbaciones menores al servicio programado [DH 2003]. En el Anexo Transporte Público se describen los problemas mencionados en los párrafos anteriores.

El proyecto PDT trata sobre la planificación estratégica, es decir, el diseño de recorridos y frecuencias. De esta manera el Proyecto de Grado queda enmarcado.

1.3 Limitaciones de los proyectos anteriores

Este proyecto debe recoger las experiencias de los Proyectos de Grado anteriores relacionados, los cuales se mencionan a continuación:

- Herramienta para el diseño de recorridos en transporte colectivo urbano. Diego Cukerman, Giovanna Garula, Matías Melgar. Año 2005. [GCM 2005]

La herramienta obtenida ayuda a la toma de decisiones para el diseño y optimización de recorridos del transporte público urbano colectivo. Asimismo incluye la implementación de mecanismos para la ejecución de algoritmos de optimización, permitiendo la visualización de soluciones para diversos casos de

estudio. También posee el diseño de un modelo de demanda integrado a la herramienta para el manejo de datos relativos al caso de estudio real de la ciudad de Rivera.

- **Interfaz Gráfica para Herramienta de Planificación de Transporte Público Urbano Colectivo.** Ana Meneghin, Andrea Ferraz-Leite. Año 2003. [MF 2003]
Este proyecto construyó una interfaz gráfica que da respuesta a la toma de decisiones durante el proceso de planificación del transporte público urbano colectivo. Además provee la invocación a un módulo de algoritmos de optimización que resuelve el diseño de recorridos y la determinación de frecuencias en un sistema de transporte público urbano.

Los productos construidos en estos proyectos presentaron limitaciones a la hora de ser implantados, ya que ambos utilizan herramientas de procesamiento de información geográfica de tipo no libre, por lo tanto generarían costos importantes al momento de conseguir la licencia por parte del cliente. Por otra parte el proyecto [MF 2003] está fuertemente acoplado a un tipo específico de algoritmos, lo cual produjo que la aplicación construida entrara en desuso debido a esa dependencia, ya que los algoritmos evolucionan rápidamente. A partir de estos elementos se considera la oportunidad de lanzar el presente Proyecto de Grado 2008 haciendo especial acento en: i) el uso de software de manejo de información geográfica de tipo no propietario, ii) el trabajo en estrecha relación al caso de estudio relativo a la ciudad de Rivera, y en el contexto del PDT, iii) restringir al mínimo posible el acoplamiento entre la herramienta y los algoritmos de forma que puedan intercambiarse de manera sencilla. En relación al punto ii), el proyecto de grado incluye también un componente de procesamiento de datos.

1.4 Objetivos del proyecto

De la sección 1.3 se desprenden los siguientes objetivos.

1.4.1 Generales.

- Desarrollo de una herramienta de apoyo a la toma de decisiones para la optimización de recorridos y frecuencias para transporte público.

- Continuar emprendimientos comenzados en Proyectos de Grado de años anteriores, como parte de una línea de investigación del Departamento de Investigación Operativa.

1.4.2 Específicos.

- Construir una herramienta capaz de representar y manipular gráficamente los recorridos, interactuar con los algoritmos de optimización y evaluación, y procesar la información necesaria para ello.
- Uso de un sistema (biblioteca de software) para el manejo de información geográfica de tipo no propietario.
- Aplicación al caso de estudio relativo a la ciudad de Rivera.

2. Desarrollo

2.1 *Requisitos Funcionales*

El propósito de este proyecto es la construcción de una herramienta con funcionalidades de interfaz gráfica para embeber un conjunto de algoritmos. El sistema a construir está dirigido a dos tipos de usuarios:

- a) diseñador de la red
- b) planificador

El usuario tipo “diseñador de la red” es aquel capaz de manejar todos los componentes de la herramienta. Es el usuario capaz de configurar los algoritmos que interactúan con el sistema. Se asume que tiene dominio sobre los modelos y algoritmos de optimización y evaluación de recorridos y frecuencias, disponibles a través de la herramienta.

El usuario tipo “planificador” es el planificador experto para el caso de estudio, encargado de la toma de decisiones acerca del mismo. Aquel que cuenta con la suficiente pericia y conocimiento local para analizar las soluciones brindadas por la herramienta, estando capacitado para editarlas bajo criterios (que le otorgan su experiencia) que crea conveniente.

De aquí en más cada vez que se mencione “usuario” se referirá a tareas que pueden ser realizadas indistintamente por ambos planificadores. En caso contrario se mencionará explícitamente el tipo de usuario.

La línea de trabajo del Departamento de Investigación Operativa, en la que se enmarca este proyecto, se enfoca en el diseño óptimo de recorridos (y sus frecuencias asociadas) en una red de transporte público (transit network design problem - TNDP). El TNDP es un problema de optimización multiobjetivo de alta complejidad combinatoria [M 2005]. Este problema motiva la necesidad de una herramienta que permita armar un caso, aplicar los algoritmos y comunicar los resultados a los usuarios. Para la construcción de esta herramienta se trabajó en estrecha relación al caso de estudio del sistema de transporte de la ciudad de Rivera.

En general los algoritmos de optimización de recorridos toman como entrada un grafo no dirigido, una matriz origen destino, las principales características de la flota de ómnibus y parámetros de diseño. El grafo no dirigido representa la red por donde

transitan los ómnibus. La matriz origen destino contiene la información de demanda entre pares de nodos del grafo. Para el modelo utilizado en el caso, la cantidad de pasajeros es la única característica de interés de los ómnibus, todos los vehículos de la flota son considerados del mismo tipo. Los parámetros de diseño pueden ser del modelo, por ejemplo demanda cubierta, o pueden ser del algoritmo, por ejemplo cantidad de iteraciones.

Una decisión importante que debe tomar el usuario al comenzar con la planificación refiere al nivel de detalle de la información que va a manejar, dado que esto impactará en la construcción de la red y la matriz origen destino. El nivel de detalle lo resuelve el usuario al momento de zonificar la región de estudio. Algunos criterios para determinar el nivel de agregación de los datos a modo de ejemplo pueden ser: barrios, paradas o micro zonas [M 2005].

El grafo representa una abstracción de la red vial. Para la construcción del grafo se debe definir sus vértices y sus aristas. Una posible forma de hacer esto último, consiste en utilizar como vértices los centroides o baricentros de las zonas y como aristas las conexiones que existen entre los centroides, las conexiones también son generadas por el usuario. Los centroides representan las zonas que visitarán los recorridos y las conexiones representan la posibilidad de que los recorridos puedan ir de una zona a otra.

Los datos de la demanda se expresan generalmente mediante una o varias matrices origen destino que expresan la cantidad de viajes que existe entre las distintas zonas de la región de estudio. En nuestro caso las distintas matrices representan la demanda en distintas horas del día. Se consideran distintos horarios dado que la variación de la demanda en matrices de un rango horario grande puede ser extensa.

La herramienta busca colaborar con el usuario al procesar la información para los algoritmos y al visualizar los resultados de éstos. También pretende facilitar al usuario la evaluación de soluciones que crea conveniente.

La aplicación construida que abarca todos los módulos se denomina igoR-tp, Interfaz Gráfica para la Optimización de Redes de Recorridos del Transporte Público. Debido a ello en las siguientes secciones se la llamará indistintamente como aplicación, herramienta o por su propio nombre. En la próxima subsección se definirán los módulos principales de igoR-tp.

2.1.1 Módulos

Se definen dos tareas elementales a ser ejecutadas por el usuario: armado de los datos y manipulación de los mismos.

La primera tarea se denominará Construcción del Caso de Estudio, y la segunda Manipulación del Caso de Estudio.

La construcción del caso de estudio es la tarea que se encarga del armado de los datos. Esto implica la construcción del grafo y la matriz de demanda asociada.

El grafo es la representación de la red vial. Para construirlo la ciudad se divide en zonas. Una zona se representa como un polígono simple. Esto quiere decir polígonos sin agujeros o islas. La herramienta facilita esta tarea. También genera de manera automática el centroide de cada zona. Los vértices del grafo están representados por los centroides y las aristas son las conexiones entre centroides. Cada arista tiene asociado un peso que se corresponde con la distancia entre los centroides. Esta distancia (distancia vial) la calcula la herramienta utilizando una implementación del algoritmo de “caminos mínimos” de Dijkstra [MS 2001]. Las figuras 2.1.1.1, 2.1.1.2, 2.1.1.3 y 2.1.1.4 ejemplifican este proceso.

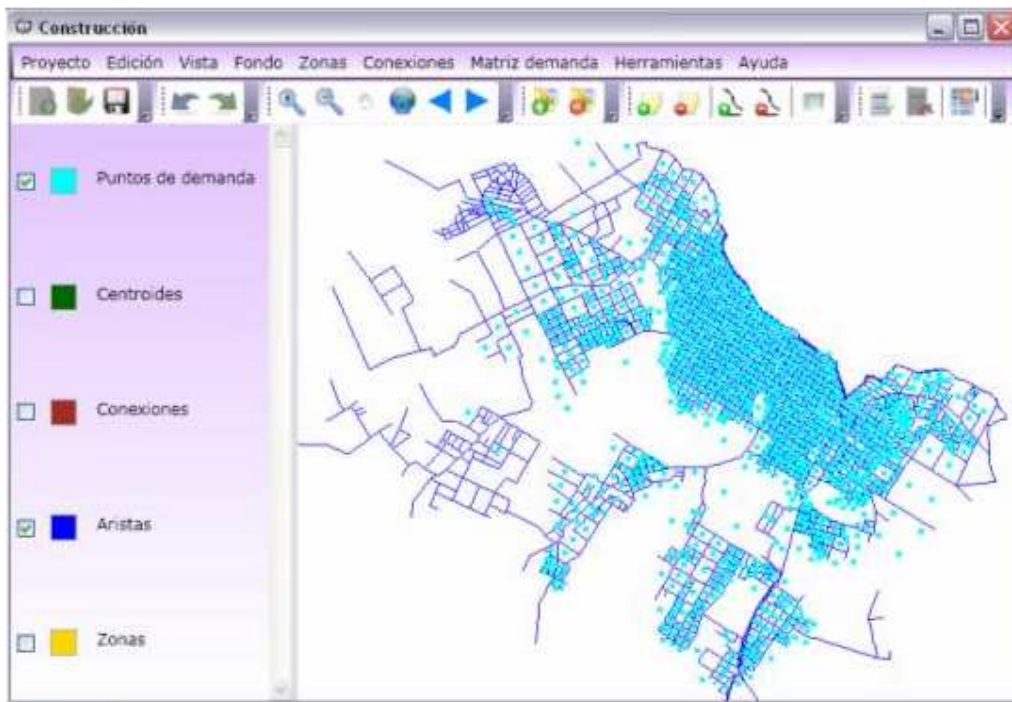


Figura 2.1.1.1. Red vial de la ciudad de Rivera.

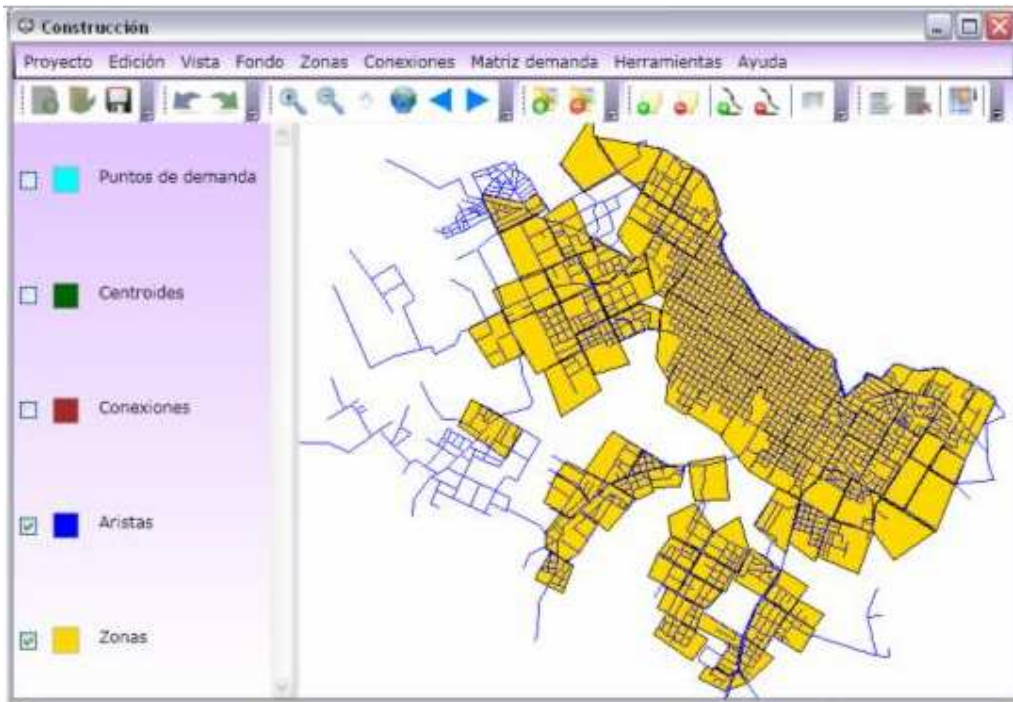


Figura 2.1.1.2. Zonificación de la ciudad.

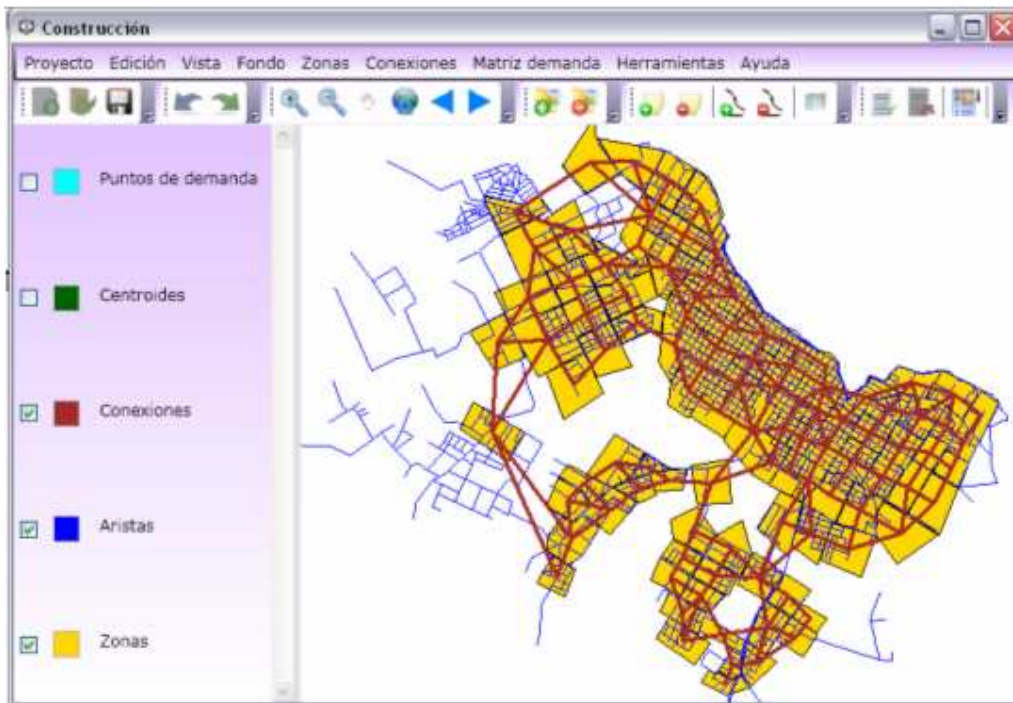


Figura 2.1.1.3. Conexiones entre las zonas generadas.

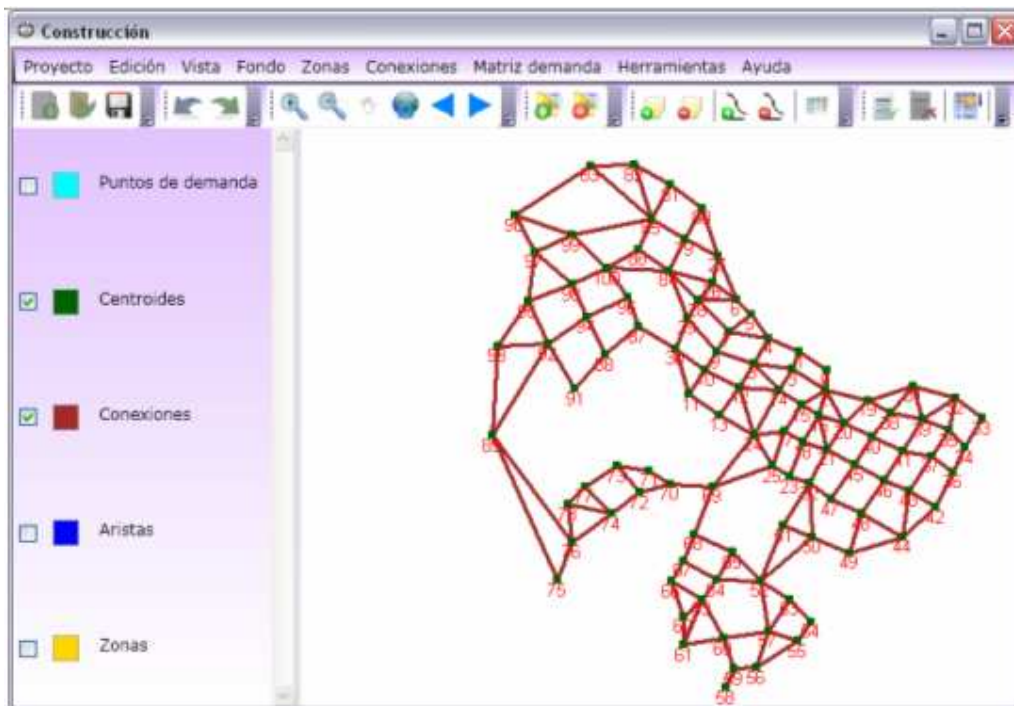


Figura 2.1.1.4. Grafo resultante del proceso de crear las zonas y conectarlas.

La demanda entre los vértices del grafo (centroides) se representa con una matriz. Para la construcción de la matriz es necesario calcular la demanda de cada zona. Los datos de la demanda deben estar asociados con algún elemento geográfico de la ciudad, por ejemplo manzanas, paradas, etc. La herramienta calcula, a partir de los datos de la demanda y los datos geográficos, la demanda entre las zonas o lo que es lo mismo entre los centroides.

El módulo de manipulación del caso de estudio se encarga de maniobrar los datos. Maniobrar los datos implica dos grandes trabajos, interactuar con los algoritmos de optimización y evaluación, y con el usuario.

Además del grafo que representa la red vial y la matriz de demanda, se necesitan las características de la flota y los parámetros de diseño.

La característica de la flota que interesa para el problema es la capacidad de pasajeros sentados de los ómnibus, además de la velocidad comercial de los mismos. Vale aclarar que la velocidad comercial es la relación que existe entre el espacio recorrido y el tiempo total de marcha entre origen y destino, más el tiempo de espera en origen [TU 2003].

Los principales parámetros de diseño para los algoritmos de acuerdo al modelo utilizado, son a modo de ejemplo, las proporciones mínimas de demanda cubierta, las frecuencias mínimas y máximas y el factor de carga máximo en los ómnibus.

Todos estos datos deben ser ingresados por el usuario antes de correr el algoritmo. Luego es necesario poder obtener las soluciones brindadas por los algoritmos de optimización, el conjunto de recorridos y frecuencias. Un recorrido es un conjunto ordenado de vértices de la red y la frecuencia es el tiempo entre pasadas de los ómnibus del recorrido. El usuario debe visualizar los datos obtenidos de la ejecución de los algoritmos y comprenderlos sin mayor dificultad. Éste además, aportará nuevas soluciones para ser evaluadas modificando o creando recorridos. Las figuras 2.1.1.5 y 2.1.1.6 muestran el grafo de entrada (generado por el proceso de construcción) al módulo de manipulación y una solución construida para este caso.

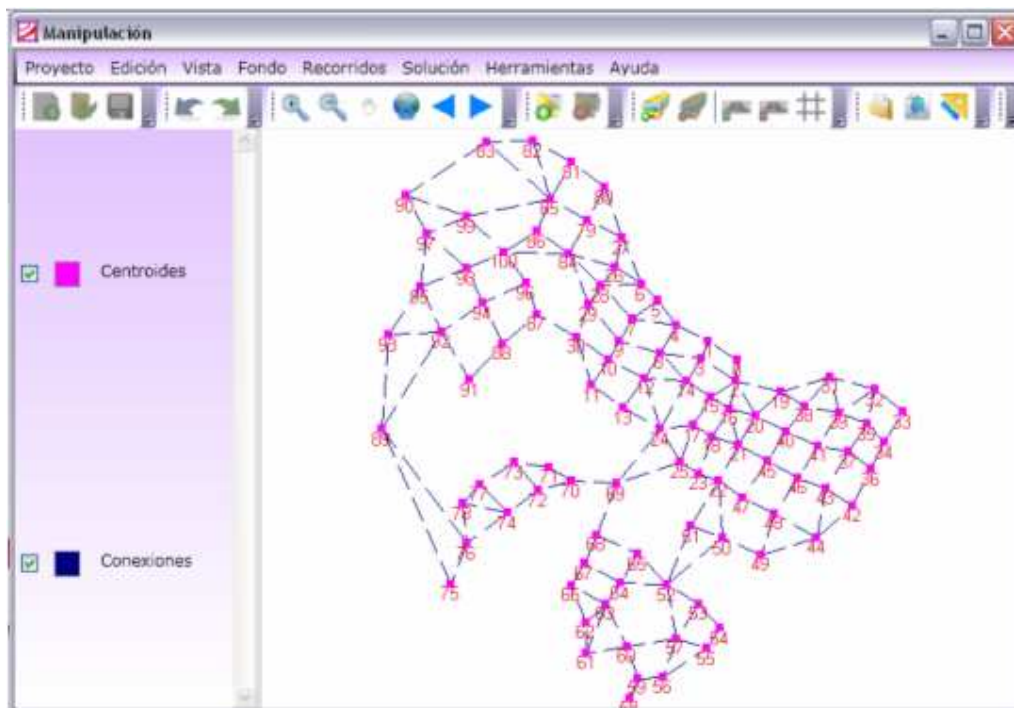


Figura 2.1.1.5. Grafo inicial, producto del proceso de construcción.

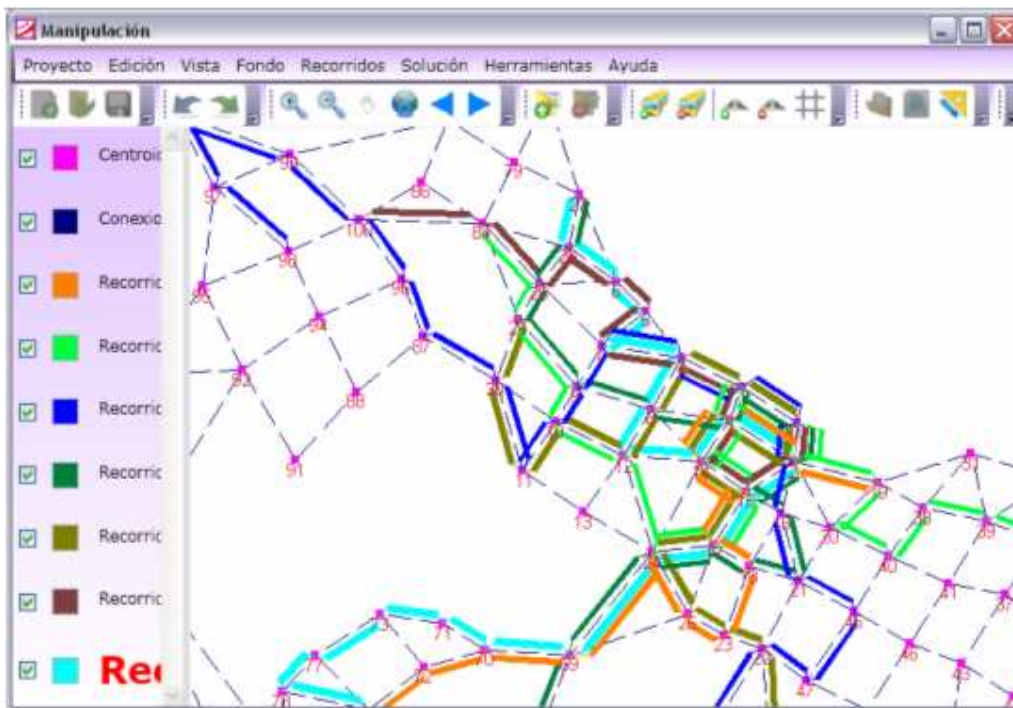


Imagen 2.1.1.6. Vista de una solución con 7 recorridos.

2.1.2 Algoritmos

El acceso a la ejecución de los algoritmos es uno de los principales requisitos de este proyecto, puesto que unos se encargan de la creación de soluciones óptimas y otros de la evaluación de soluciones. Como se puede apreciar en el manual de usuario en la sección 5, los mismos interactúan con la herramienta mediante archivos en formato de texto.

Por lo tanto los algoritmos reciben a través de igoR-tp: grafos que indican las aristas que unen los distintos centroides con sus respectivos pesos, rangos de frecuencias y la matriz de demanda que contiene todos los centroides origen-destino entre los cuales existe demanda. Una vez procesada esta información y ejecutado el algoritmo, este devuelve otro archivo en formato “txt”, con las soluciones propuestas, en definitiva un conjunto de recorridos y sus respectivas frecuencias.

Una etapa previa es determinar qué algoritmo se desea utilizar, para lo cual igoR-tp cuenta con un módulo que permite dar de alta archivos de configuración, los cuales determinarán qué algoritmos pueden ser empleados. Más información acerca del funcionamiento de este módulo se encuentra en el manual de usuario en la sección 4.3.1

(Crear archivos de configuración). Las figuras que aparecen a continuación permiten visualizar el proceso de creación y utilización de los algoritmos. En la figura 2.1.2.1 se muestra la creación de un archivo de configuración para un algoritmo que requiere de tres parámetros de entrada. En la figura 2.1.2.2 se aprecia como se asignan valores a esos parámetros y en la figura 2.1.2.3 se puede ver el archivo que se genera a nivel interno para manejar la información de los algoritmos. Cabe mencionar que el usuario no manipula este archivo directamente, simplemente cuando desee trabajar con algoritmos lo puede hacer mediante las interfaces que presenta la herramienta, como se observa en las figuras 2.1.2.1 y 2.1.2.2.



Figura 2.1.2.1 Creación de archivo de configuración de un algoritmo con tres parámetros.

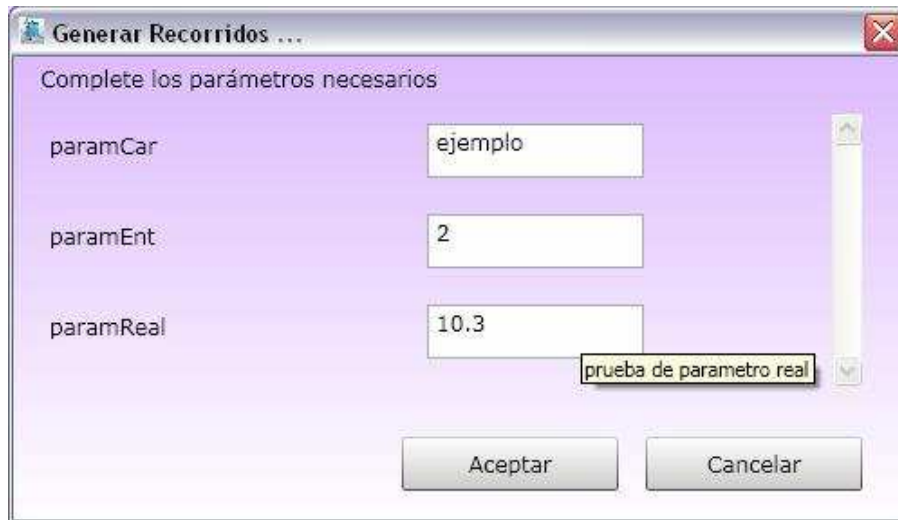


Figura 2.1.2.2 Asignación de valores para los parámetros de un algoritmo.

```
<?xml version="1.0" standalone="no" ?>
<!--Archivo de configuracion del algoritmo: algoritmoTest-->
<Algoritmo>
  <nombre>OptimEx.exe</nombre>
  <ruta>C:\Program Files\igoR-tp v1.0\OptimEx</ruta>
  <parametros>
    <parametro>
      <nombre>paramCar</nombre>
      <tipo>caracteres</tipo>
      <observacion>prueba de parametro caracteres</observacion>
    </parametro>
    <parametro>
      <nombre>paramEnt</nombre>
      <tipo>entero</tipo>
      <observacion>prueba de parametro enteros</observacion>
    </parametro>
    <parametro>
      <nombre>paramReal</nombre>
      <tipo>real</tipo>
      <observacion>prueba de parametro real</observacion>
    </parametro>
  </parametros>
</Algoritmo>
```

Figura 2.1.2.3 Archivo de configuración del algoritmo Optimex.

El archivo de configuración de los algoritmos contiene el nombre del ejecutable del algoritmo y su ubicación. Además cuenta con información de cada uno de los parámetros: nombre, tipo y observación. Luego este archivo será consumido por el módulo de manipulación del caso de estudio para generar soluciones o evaluarlas utilizando el algoritmo de optimización o de evaluación (Optimex.exe en el ejemplo).

Los parámetros que requiere el algoritmo serán completados por el usuario y enviados de tal forma que posteriormente el ejecutable del algoritmo creará la salida con el formato necesario para que el módulo de manipulación la reconozca.

De acuerdo a lo mencionado en esta sección se nota que la herramienta desconoce la estructura de los algoritmos, es decir son independientes salvo por el protocolo de comunicación. Esta propiedad fue uno de los requisitos exigidos, puesto que los algoritmos de optimización pueden cambiar. Por lo tanto si la herramienta y los algoritmos fueran componentes dependientes, esto los ligaría de tal manera que igoR-tp, entraría rápidamente en desuso. Con lo cual se puede predecir que la independencia lograda, servirá para interactuar con distintos tipos de algoritmos a medida que estos cambien.

2.2 Requisitos no Funcionales

Los requisitos no funcionales de la aplicación son los siguientes: a) utilidad, c) documentación, c) herramienta SIG de tipo no propietario.

La utilidad es una cualidad que toda herramienta aspira. Es una cualidad muy importante para todo software y fundamental para una interfaz gráfica. Sobre este requisito y cómo fue abordado el mismo, se encuentra en detalle en la sección 2.4 Interacción herramienta usuario.

La documentación consiste fundamentalmente en dos documentos: uno técnico que especifica las funcionalidades que brinda la herramienta, y un completo manual de usuario que describe como utilizar las funcionalidades de igoR-tp.

Esta sección se concentrará en el tercer requisito no funcional, herramienta SIG de tipo no propietario.

Un SIG (Sistema de Información Geográfica) es un sistema de computación capaz de crear, visualizar y manipular información que posee características geográficas [EGG 2008]. Con este tipo de herramienta uno logra interpretar de forma más clara y concisa los datos, denotándose las relaciones y patrones existentes entre ellos.

Debido a que la información que maneja la aplicación a construir en el presente proyecto tiene fuertes componentes geográficos, es imprescindible que cuente con una herramienta de manipulación de información geográfica que ayude a procesarla.

La herramienta SIG debe ser tipo no propietaria. El motivo principal es el alto costo que tienen las herramientas SIG propietarias, que hacen que sean inaccesibles para este proyecto. Además, no requiere hacer uso de funcionalidades complejas de SIG, lo que no justifica el gasto, aún si se tuvieran los recursos.

Para poder seleccionar una herramienta, primero se analizó qué características eran deseables que cumpliera. Se necesitaba que fuera posible un completo manejo del formato ESRI Shapefile, esto incluye creación, consulta y visualización de capas y además que fuese posible agregar y eliminar elementos de las mismas. Se escogió el formato ESRI Shapefile dado que es un formato que la mayoría de las herramientas SIG lo manipulan, es un estándar de facto. Por otra parte los datos que se tienen para el caso de estudio están en este formato.

Se llevó a cabo una investigación entre varias herramientas SIG de tipo no propietaria que culminó con la selección de la herramienta MapWinGIS; la descripción de las distintas opciones así como de los criterios y razones de la elección se encuentran en el Anexo Búsqueda y Selección de Tecnologías SIG.

Para la construcción de la herramienta igoR-tp se utilizó de MapWinGIS las funcionalidades: manejo del formato ESRI Shapefile, generación de centroides a partir de polígonos, y funciones de recorte de polígonos.

2.3 Diseño y Arquitectura

La herramienta se compone de tres módulos, más un componente madre que los centraliza. La mayor parte de la lógica de todo el producto se encuentra en los módulos de construcción de caso de estudio (1) y de manipulación del caso de estudio (2). En ellos las características del diseño son similares, identificándose dos elementos principalmente:

- Un componente para la manipulación de la información geográfica. Se organiza en forma de biblioteca brindando sus funcionalidades al otro componente.
- Una columna vertebral diseñada en base al modelo de capas. Este componente es el que interactúa con el usuario (capa presentación), intercambia información con el componente geográfico, posee toda la lógica del modelo de dominio (capa lógica), además del acceso a datos y persistencia de los mismos (capa de persistencia).

El modelo de capas fue elegido por contar con las siguientes características: Hay varias capas dónde las inferiores brindan servicios a las superiores. Además el acceso a servicios de otras capas está limitado a la capa inmediata inferior. Este modelo proporciona ventajas tales como facilitar la comprensión, ya que se basa en niveles de abstracción; facilitar el mantenimiento, porque es posible modificar una sola capa sin afectar al resto; facilitar la reutilización y portabilidad [ABDLFPTVW 2008].

El modelo en capas presentó dificultades al momento de diseñar la integración con el componente de manipulación de información geográfica. Por ejemplo, determinar si una sola capa se comunicaría o si el acceso sería a través de cualquiera de ellas, este problema fue posible solucionarlo en forma consistente accediendo sólo a través de capa lógica. Este modelo puede deteriorar el desempeño al existir tiempo consumido en la coordinación de las distintas capas, pero por no ser una herramienta en la que el factor del tiempo sea sumamente prioritario, no pareció un obstáculo de gran peso para desecharlo.

Por lo tanto se optó que las propiedades del modelo de capas eran las más indicadas para el desarrollo que se llevaría adelante.

A continuación se muestran las figuras 2.3.1 y 2.3.2, que denotan la interacción entre los distintos componentes del sistema.

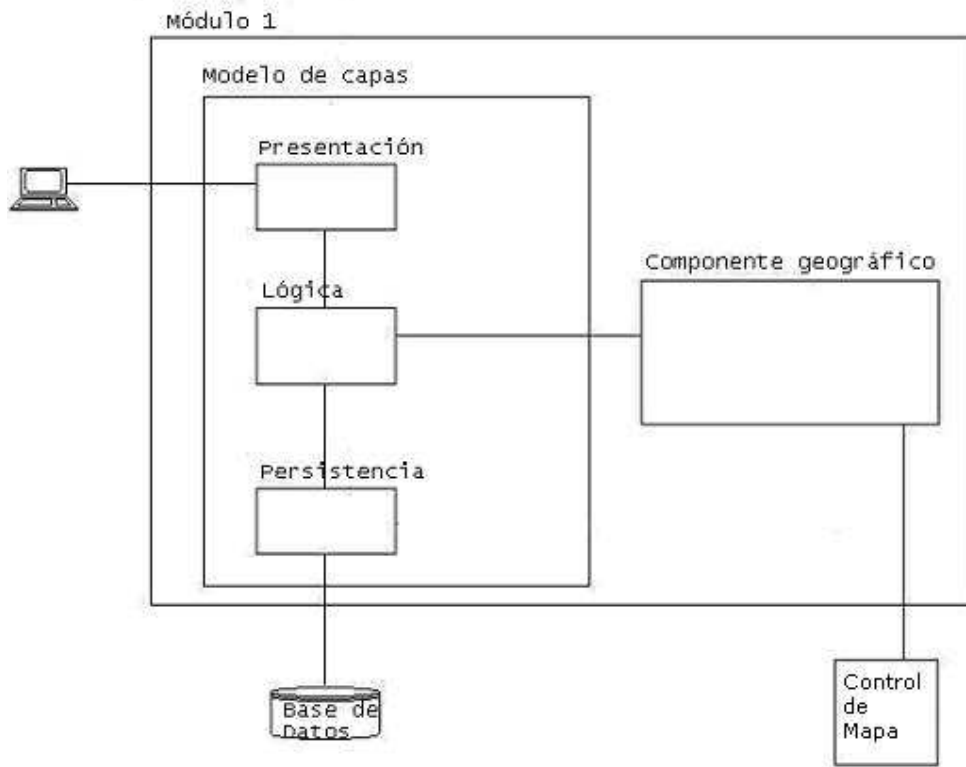


Figura 2.3.1

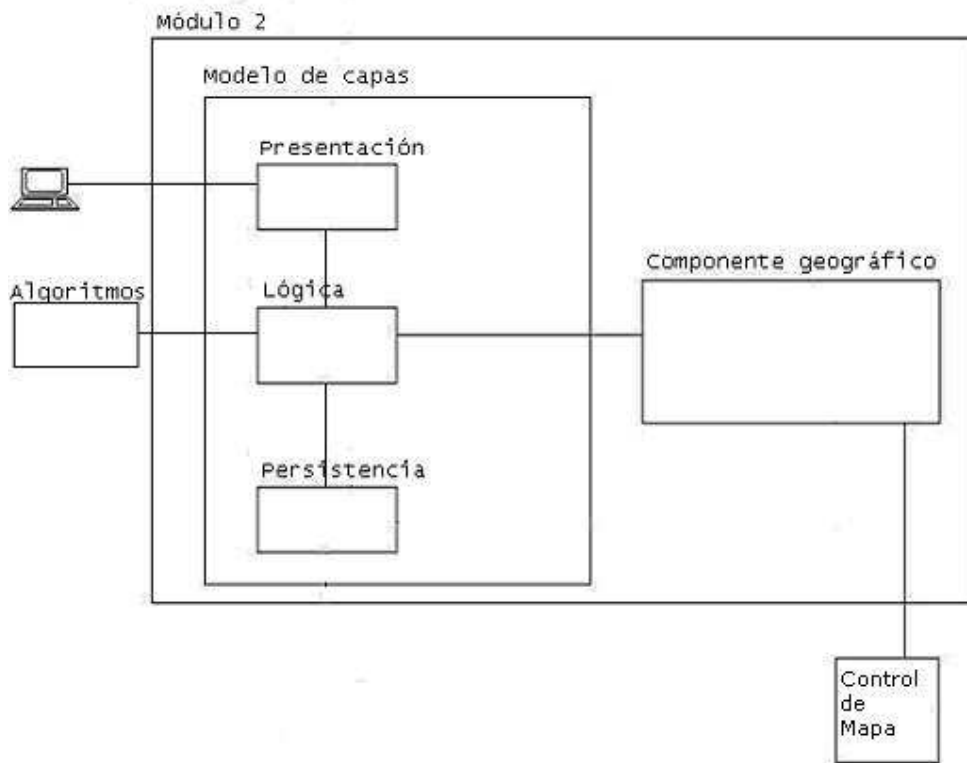


Figura 2.3.2.

Los otros módulos, el de creación y eliminación de archivos de configuración para los algoritmos del módulo de manipulación y el de centralización de toda la herramienta, cuentan con un diseño simple cuyas funciones son específicas. Los mismos surgieron como consecuencia de facilitarle al usuario todas las funcionalidades necesarias para un uso más cómodo y centralizado de la herramienta, además de entregarle al usuario una sensación de integridad de todos los componentes.

2.4 Interacción con el Usuario

Debido a que el producto resultado de este proyecto es una interfaz gráfica, fue necesario hacer especial hincapié en su interacción con los usuarios.

Si bien el sistema será utilizado por diseñadores de redes acostumbrados al manejo de este tipo de herramientas, los cuales cuentan con un perfil que les permite manipular directamente los algoritmos de optimización, sería interesante la utilización del sistema por parte de usuarios con un perfil de planificadores.

Considerando esta situación es probable que existan problemas al momento de implantación del sistema. Como mencionan G.W. Dickson y J.K. Simmons [M 2008] algunos de estos aspectos podrían ser:

- romper con los límites de departamentos o secciones establecidos
 - por ejemplo, podría provocar que para un mejor aprovechamiento del sistema dos departamentos o secciones se fusionen (planeamiento y tránsito).
- romper la red de comunicaciones informales
 - la comunicación informal se genera en forma espontánea sin ningún tipo de protocolo.
- exigir características individuales específicas
 - las personas con mucha antigüedad en sus cargos ante el planteo de nuevos desafíos pueden ofrecer resistencia al cambio.
- apoyo de la organización cultural
 - la alta dirección debe apoyar la implantación si se quiere que sea exitosa.
- opinión de los usuarios en cuanto a la manera de implantación.
 - integrar administradores y subordinados para la toma de decisiones produce mayores probabilidades de éxito.

Las formas de rechazo se pueden traducir en agresión (sabotaje al sistema), proyección (se le atribuye los problemas siempre al sistema), prevención (se ignoran resultados obtenidos por sistema).

Luego de revisar los obstáculos que podrían aparecer al implantar el sistema, se realiza a partir de los mismos el valor de la interacción usuario – sistema. Mediante esta interacción se puede ayudar a la superación de estos potenciales problemas. Por ejemplo si se consiguen las siguientes características para el sistema producido:

- Amigable
- Facilidad de uso
- Diseño centrado en el usuario

Estos conceptos permitirían que el aprendizaje para el uso del sistema fuera más corto. Además de reducir la oposición al uso del mismo por brindarle a los usuarios, lo que ellos necesitan de manera sencilla. En definitiva conseguir una buena interacción significaría una gran ventaja para el éxito posterior a la implantación.

La interfaz es todo lo que el usuario ve del sistema. Este punto es una muestra clara de la importancia del aspecto de la herramienta para la persona que la utiliza. A pesar de que se logre construir un producto con muchas virtudes como ser una codificación óptima y una buena arquitectura, si el aspecto de interacción no es tenido en cuenta al momento del diseño, de poco servirán las otras virtudes. El diseño debe servir al usuario. Tomando esta noción como puntapié inicial se ha apostado a conseguir un producto con las características mencionadas unas líneas atrás. A continuación se definen de forma más específica.

- Amigable: “Dícese de la interfaz de usuario basada en elementos y metáforas gráficas que facilita la interacción con el ordenador y con las aplicaciones que corren en éste.”

Extraído de: [MM 2004]

- Facilidad de uso: "La facilidad de uso se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso " [ISO/IEC 9126]

"La facilidad de uso es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico" [ISO/IEC 9241]

Extraído de: [M 2002]

- Diseño centrado en el usuario: uno de los objetivos esenciales para conseguir una buena interacción es conocer quienes serán los usuarios y sus intereses para poder satisfacerlos.

El diseño centrado en el usuario fue el hilo conductor de nuestra tarea. Mediante el relevamiento de requisitos se conoció lo que la herramienta debía hacer. Esto generó un diseño para cumplir con los requisitos especificados. Al completar estas etapas se trabajó primero con prototipos en papel y luego en computadora validando las sucesivas versiones con el propio usuario (Anexo Prototipos). De esta manera se recibió constante retroalimentación del diseñador de la red acercándonos cada vez más a lo que se esperaba de la herramienta.

Para poder obtener un producto fácil de usar y amigable se analizaron algunos principios para el diseño de interfaces de usuario, los que intentamos seguir de la mejor

forma posible, ya que su efectividad es estudiada en el área de interacción persona-computadora.

Estos principios son:

1. Organizar: implica brindarle al usuario una interfaz clara y de consistente estructura conceptual.
2. Economizar: obtener una gran efectividad a partir de un conjunto pequeño de indicadores.
3. Comunicar: presentar los elementos para que el usuario sea capaz de comprenderlos por sus propios medios.

En las próximas subsecciones se describirán estos principios y como fueron aplicados en igoR-tp.

2.4.1 Organizar.

Según Josef Albers – artista de la Bauhaus, “Diseñar es planificar y organizar, ordenar, relacionar y controlar. En breve abarca todos los medios para oponerse al desorden y a los accidentes. Por tanto, es una necesidad humana que califica el pensamiento y la acción de las personas.”

Con estas palabras se evidencia la importancia del diseño en distintos ámbitos, y las ventajas que se desprenden si se lo aprovecha de manera adecuada. Este principio se puede notar en las siguientes imágenes.

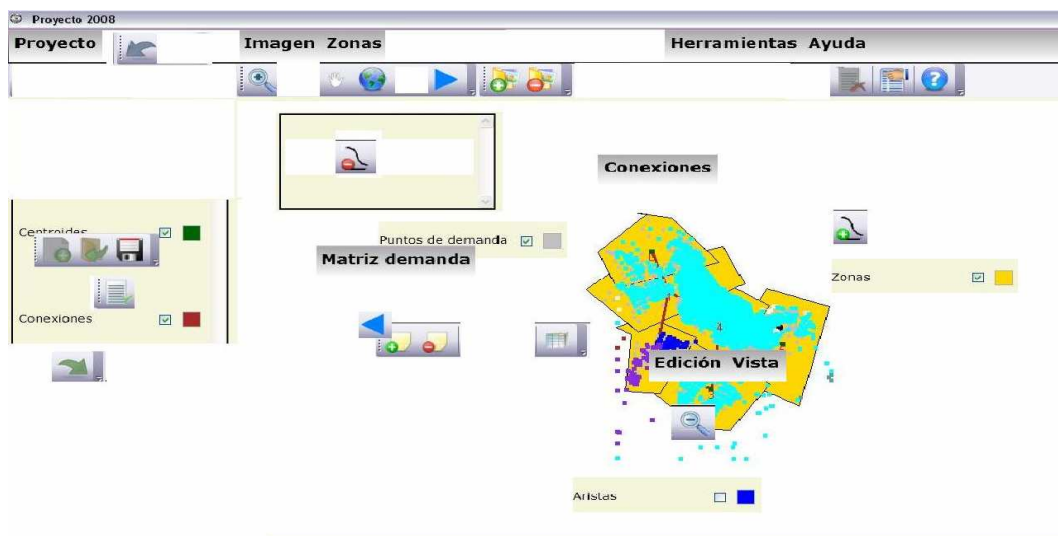


Figura 2.4.1. Imagen de diseño tipo caótico.

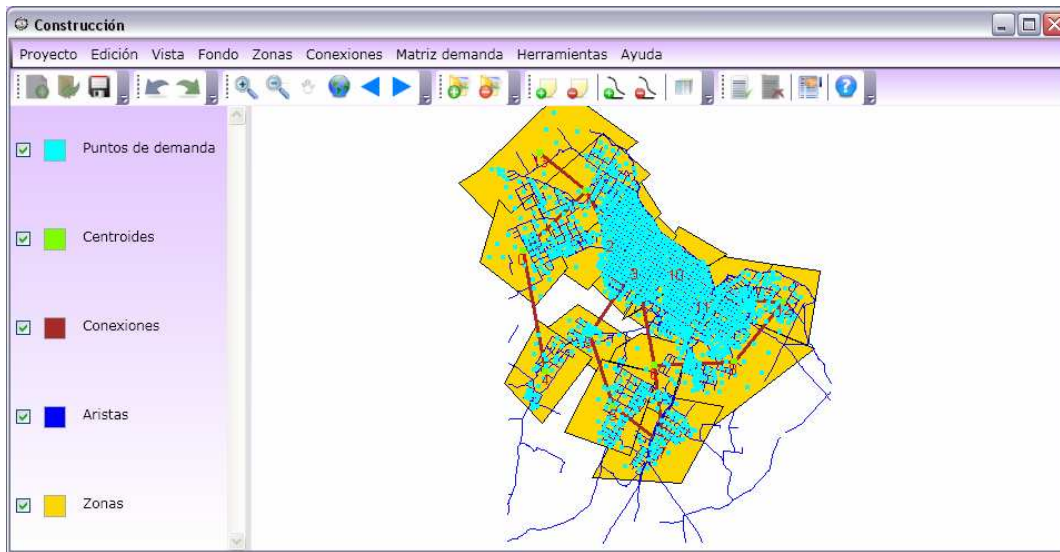


Figura 2.4.2. Imagen del módulo de construcción.

La consistencia de la herramienta tiene dos ramas.

Consistencia interna: se buscó estandarizar todos los elementos en las distintas secciones que componen la interfaz gráfica. Por ejemplo se utilizó el mismo tipo de fuente, para las etiquetas, los menús, los botones, las grillas. El tamaño de los botones es siempre igual y además reaccionan de la misma forma en cuanto a su aspecto, tanto al pasar el cursor del ratón sobre ellos como al seleccionarlos. La disposición de las grillas es similar en todos los casos.

Las observaciones anteriores son ciertas para la mayoría de los casos, salvo para aquellos elementos empleados que dependen del sistema operativo, un claro caso de ello es el tipo de fuente que se ve en los títulos de las ventanas y el propio contorno de las mismas.

Consistencia externa: se pretendió utilizar convenciones de otras aplicaciones del mismo tipo como son las herramientas SIG, intentando que el usuario ya esté familiarizado con su aspecto. A modo de ejemplo se pueden notar las similitudes entre la Figura 2.4.3 ArcExplorer 4.0.1 y la Figura 2.4.2. Este punto también abarca convenciones culturales, como ser el lenguaje que maneja la herramienta en su interfaz gráfica. Mayoritariamente se utiliza el idioma español, ya que el idioma inglés podría ser un obstáculo para algún usuario.

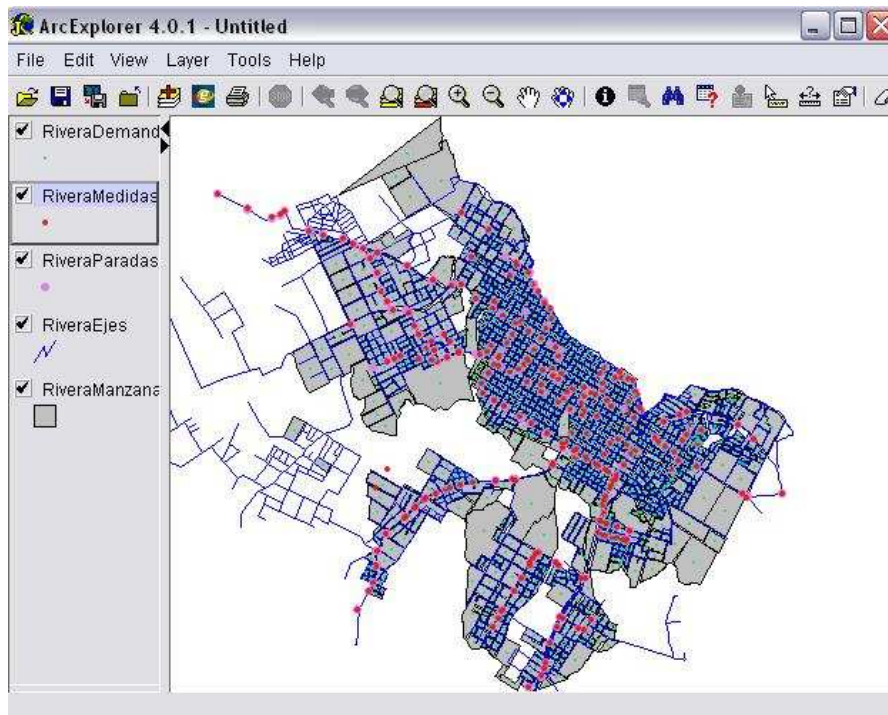


Figura 2.4.3 ArcExplorer 4.0.1

Para la disposición de las ventanas que componen la herramienta se cuidaron tres formas principales de organización: se utilizaron estructuras de grilla, se estandarizó la disposición de las mismas y se agruparon elementos relacionados. Esto se puede ver en la siguiente imagen (Figura 2.4.4). En la misma también se aprecia que el número de grandes divisiones cumple la regla de Miller, el número mágico 7 ± 2 [M 2001]. Este número indica las limitaciones que presenta nuestro sistema de recepción para percibir información y memorizarla (una persona es capaz de retener entre 5 y 9 elementos con su memoria de corto plazo). Esta condicionante fue trabajada a nivel de la interfaz, ya que no está basada en la memoria, además se utilizaron íconos, ventanas y menús para agrupar elementos y permitir una identificación de los mismos con mayor facilidad.

Por otra parte se intentó establecer una clara relación entre los elementos, vinculando y desvinculando según su similitud funcional (Ver figura 2.4.4).

Asimismo se tuvo en cuenta la navegabilidad en cada ventana, proveyéndole un foco inicial a la atención del usuario, dirigiendo la atención hacia donde parecía más importante de acuerdo al comportamiento esperado del usuario. (Ver figura 2.4.4)

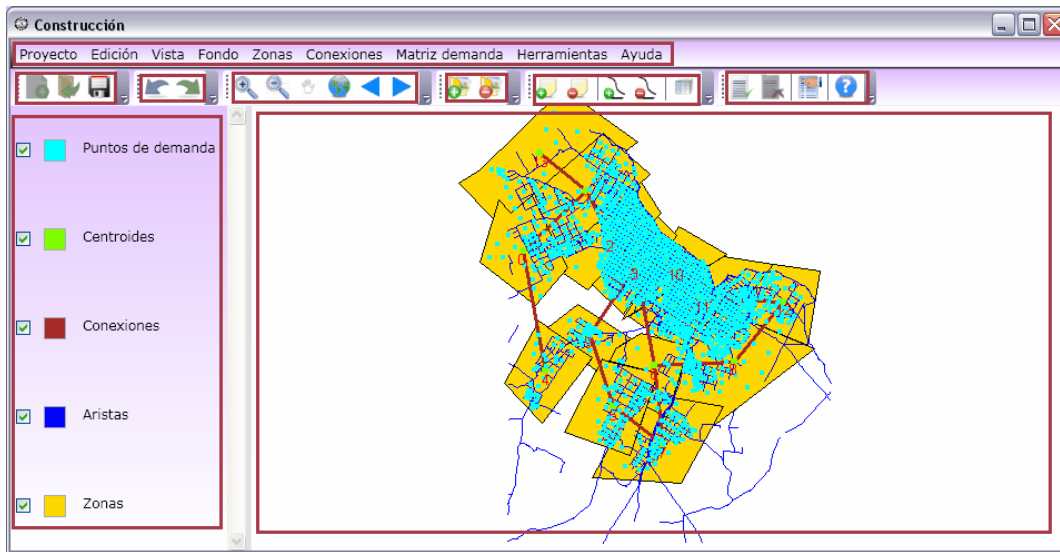


Figura 2.4.4.

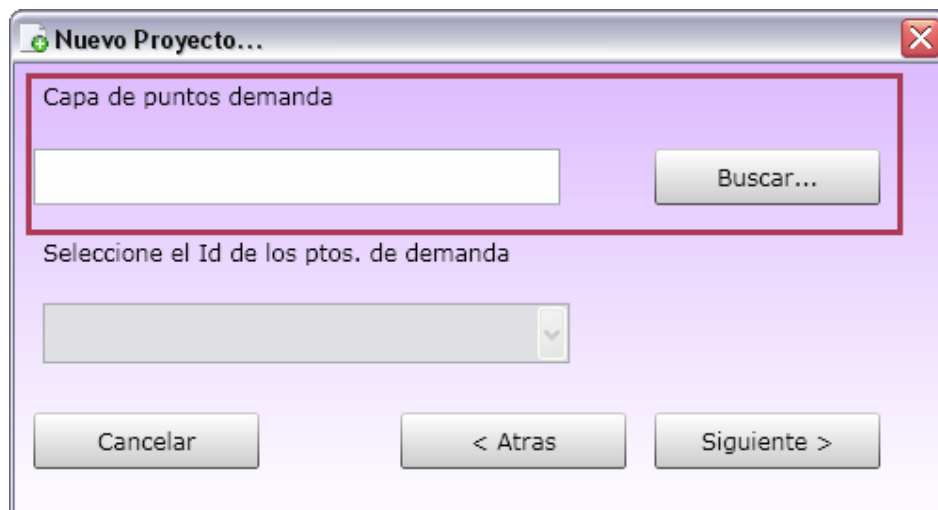


Figura 2.4.5.

2.4.2 Economizar.

“Para que algo luzca fácil, minimiza el número de controles”. Don Norman [FL 2007]. La frase anterior define este principio, en donde se muestra cómo generar un diseño sencillo (económico). Aquí se desprenden cuatro temas: simplicidad, claridad, grado de distinción y énfasis.

La simplicidad es una virtud, por lo tanto se mantuvo la interfaz lo más simple posible, incluyendo sólo los elementos que son esenciales para la comunicación. Hipotéticamente el usuario no debe ser conciente del trabajo que realiza la interfaz gráfica para transmitir lo que se desea. Esta idea se refleja con las palabras adoptadas por Ludwig Mies Van Der Rohe, “Less is more” [FL 2007]. Menos es más, mostrando la importancia de la sencillez en el diseño para comunicarse mejor con el usuario. (La figura 2.4.6, complicaría el diseño innecesariamente)

La claridad de los componentes fue tomada en cuenta para que su significado no sea ambiguo y se produzcan comportamientos inesperados para el usuario (Figura 2.4.7).

Los otros temas que consideramos son el énfasis y el grado de distinción, es decir distinguimos las propiedades importantes de los elementos esenciales, para que estos fueran fácilmente percibidos, en nuestra aplicación esto se denota con el despliegue de las capas y sus propiedades. (Figura 2.4.8)

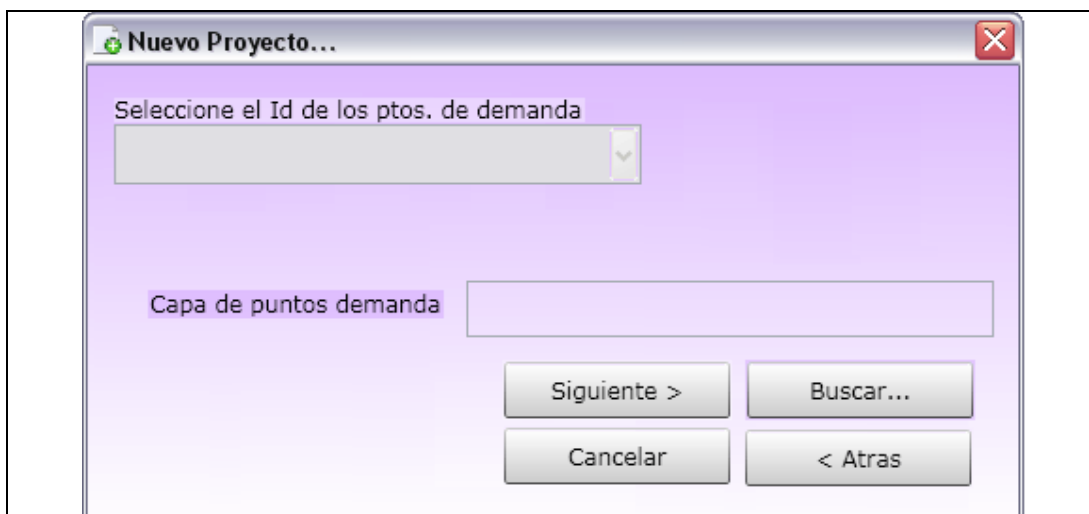


Figura 2.4.6.



Figura 2.4.7. Íconos que permiten acercarse y alejarse del mapa. (Zoom In-Out)

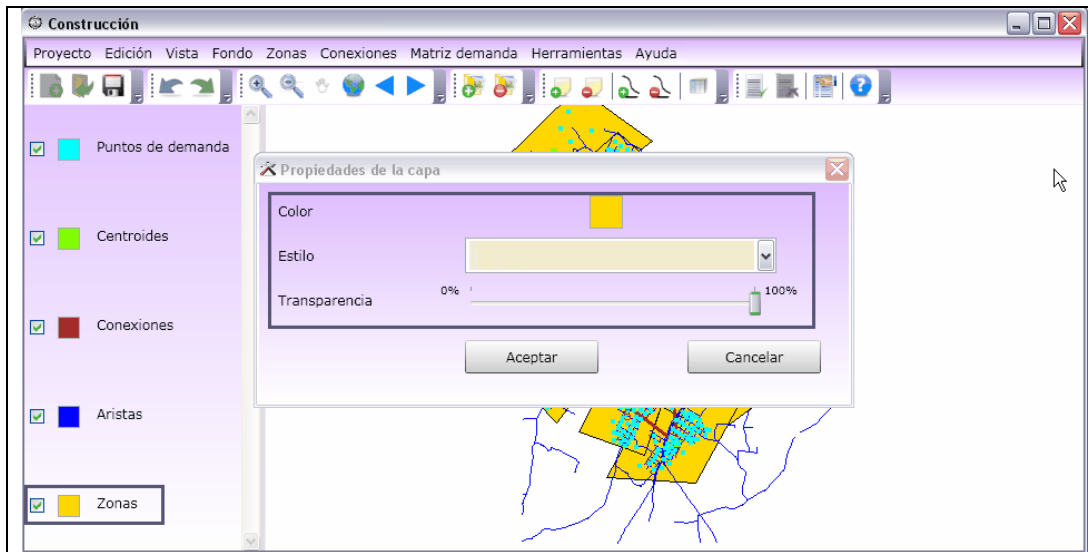


Figura 2.4.8.

2.4.3 Comunicar

“La comunicación es un proceso social dentro de un contexto específico, en el cual signos son producidos y transmitidos, percibidos y tratados como mensajes en los cuales el significado puede ser inferido”. Sol Worth [FL 2007].

Para la herramienta construida la comunicación se encuentra en diversos factores, como son:

- La legibilidad: las letras, números, símbolos y elementos gráficos son fácilmente distinguibles. El estilo utilizado en las fuentes de todas las interfaces y menús es de un tipo que no tiene ‘serif’ (no son redondeadas) evitando este hecho el problema del ‘aliasing’ de los monitores, lo cual genera mayor dificultad para la lectura de los elementos.
- La comprensibilidad: los elementos desplegados son fáciles de identificar e interpretar. Este factor es ayudado ya que se usó un solo tipo de fuente para prácticamente toda la aplicación, teniendo en cuenta de esta manera la tipografía. Intentamos no sobrecargar las funcionalidades, por ejemplo en los botones de la barra de herramientas, ya que cada uno realiza una

acción específica, con el fin de no confundir al usuario y que este sepa exactamente la opción a seleccionar de acuerdo a lo que desee hacer.

(Ver figura 2.4.9).



Figura 2.4.9. Íconos para activar y desactivar las notas del mapa.

- Simbolismo: Los símbolos e íconos incluidos en el programa fueron cuidadosamente contruidos, para transmitir el significado deseado.
- El color: es un elemento muy utilizado en toda la herramienta para realzar la información que deseamos desplegar. Este hecho se nota claramente en los íconos, los cuales tienen color cuando están habilitados para ser seleccionados o en blanco y negro en caso contrario. Por otra parte el color se puede usar para distinguir mejor las distintas capas que están cargadas en la herramienta, facilitando así su mejor reconocimiento.

Una vez que se han visto los principios para el diseño de una interfaz gráfica aplicados a igoR-tp, nos concentraremos en otros puntos importantes en cuanto a la comprensión y la facilidad de uso de la herramienta.

Existen cuatro principios según el autor Donald Norman [FL 2007], ellos son:

- La visibilidad: el usuario sabe en qué estado se encuentra actualmente y qué acciones tiene permitidas.
- Un buen modelo conceptual: la herramienta presenta las funciones posibles y sus respectivas respuestas en forma consistente.
- Un buen mapeo: la herramienta nos permite predecir los efectos de las acciones que realicemos. En este principio se desprenden otras ideas como ser las “abordabilidades” (affordances), puesto que los controles son fácilmente percibidos e interpretados, además las acciones pueden ser descubiertas, ya que se cumplieron estándares de aplicaciones del mismo estilo. A modo de ejemplo,

la forma en que se despliegan las capas que se encuentran cargadas en la herramienta.

- Retroalimentación (Feedback): el usuario recibe en todo momento información de lo que la herramienta está haciendo.

(Ver figura 2.4.10 y 2.4.11. Elementos inactivos están en blanco y negro)



Figura 2.4.10.

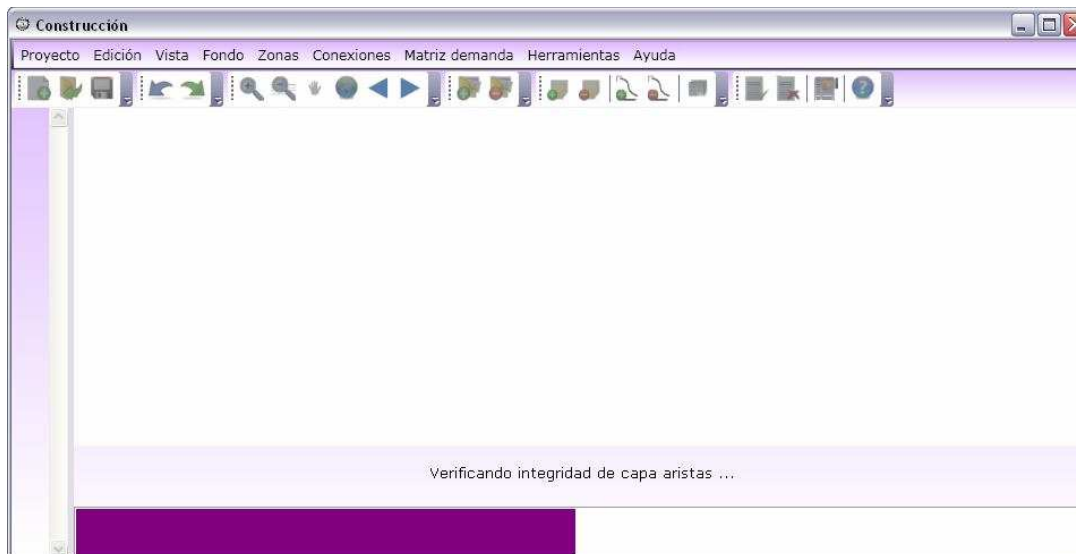


Figura 2.4.11. Sistema indicando que está haciendo.

Estos principios han colaborado también con un diseño para evitar errores; por ejemplo acceder a funcionalidades incorrectas en el caso que todas estuvieran habilitadas en todo momento. Por otra parte se tuvo en cuenta algunos criterios para ayudar a los nuevos usuarios de la herramienta, para que comprendan que es lo que pueden hacer con ella. Debido a ello, como se mencionó anteriormente seguimos el uso de estándares de herramientas del tipo SIG, usando palabras para describir a los objetos percibidos y metáforas como los botones y los propios elementos de la barra de herramientas.

Para respetar los principios de visibilidad y buen mapeo la interfaz trabaja en forma similar a una máquina de estados, brindándole al usuario una constante retroalimentación del estado del sistema, como se muestra en el figura 2.4.12:

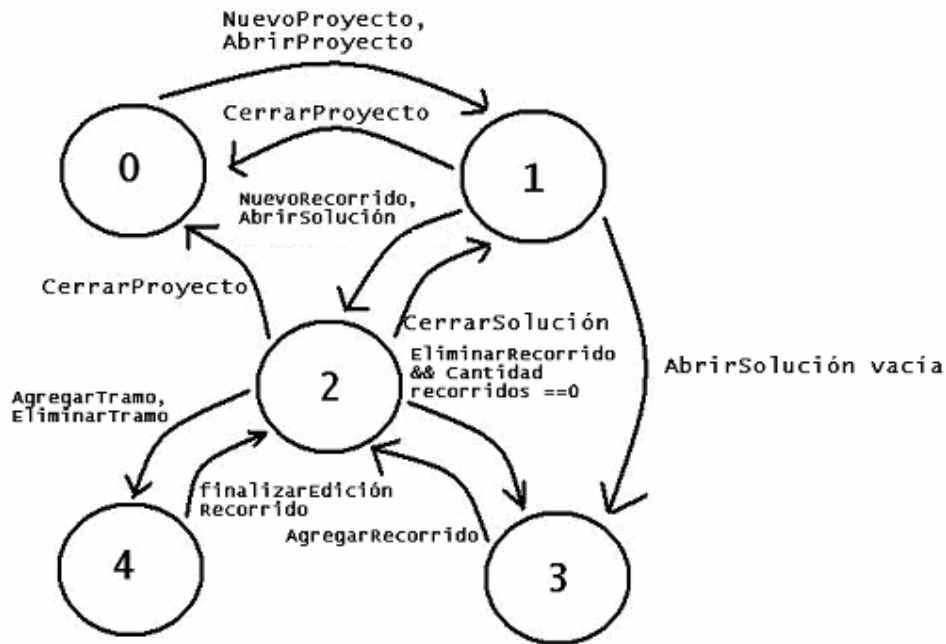


Figura 2.4.12. Estados del módulo de manipulación. (módulo de construcción análogo)

Otro concepto que no se quiso pasar por alto es el de Manipulación Directa, que se define mediante su principal objetivo que indica, que es conveniente realizar el manejo de una herramienta computacional abstracta mediante una interfaz gráfica completamente, tal que exista conexión con la manera de pensar del usuario. Esto se podría reflejar en que la herramienta igoR-tp, intenta cubrir todos los aspectos para que el usuario pueda controlar las distintas situaciones siempre a través de su interfaz gráfica, un ejemplo de ello es la generación de los archivos de configuración para los algoritmos.

Algunas virtudes de la manipulación directa se pueden observar en el producto como ser una retroalimentación continua y visual, teniendo en cuenta un interés mayor por el tiempo percibido que el real, denotándose ello en alguna de las situaciones de espera, en donde la barra de progreso se llena en sucesivas ocasiones hasta completar la tarea, en lugar de hacerlo una sola vez, de manera más lenta. Asimismo la herramienta da la posibilidad al usuario de que la mayoría de sus acciones sobre el mapa sean reversibles, pudiendo “deshacer” alguna de ellas para cambiar la dirección de su actividad.

La eficiencia de la interfaz fue tenida en cuenta, la misma se basa en que el usuario no deba entregar información si esta es innecesaria para que el sistema siga trabajando. Un ejemplo de ello, son las notificaciones que se presentan al abrir o generar un nuevo proyecto, las cuales se cerrarán solas salvo que el usuario desee hacerlo antes.

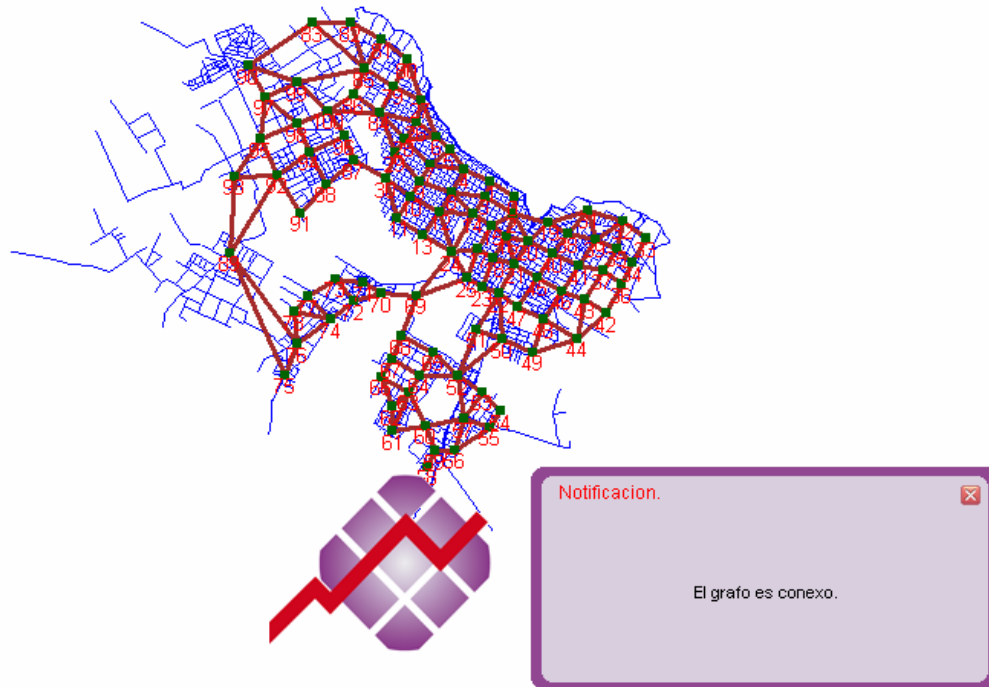


Figura 2.4.13.

Esta sección se ha concentrado en las distintas características que presenta igoR-tp con respecto a su interfaz gráfica en cuanto a la interacción con el usuario. La facilidad en el uso de cualquier herramienta de software comienza desde su instalación, para lo cual nuestra aplicación cuenta con un instalador que facilita esta tarea.

2.5 *Producto final*

Es difícil determinar realmente cuando un producto es finalizado. Nosotros tomamos esta decisión basándonos en criterios, como ser el cumplimiento de los

requisitos planteados por el usuario en la etapa inicial, conformidad del mismo con el producto resultante y estabilidad conseguida con relación a las fallas del sistema.

En el proceso de construcción del producto se presentaron varios desafíos que fueron enfrentados mediante decisiones de diseño, para solucionarlos y lograr una mejor adecuación al uso, según las necesidades del cliente. Aquí se muestran algunos ejemplos de estos retos y en el Anexo Caso de Estudio: Ciudad de Rivera, se detalla un caso típico de la utilización de igoR-tp.

- *Superposición de zonas.* Este tema genera problemas a la hora de saber que puntos de demanda incluye cada zona, a simple vista observando el mapa. Por lo tanto se optó porque no existan zonas superpuestas. Esto se logró utilizando las operaciones de recorte de polígonos de la herramienta geográfica. De esta manera si el usuario crea una nueva zona que se superpone con zonas previamente creadas, la zona resultante es el polígono diferencia entre la nueva zona y las zonas viejas. (Figura 2.5.1, 2.5.2)
- *Visualización de la información de la matriz de demanda origen-destino en el mapa.* Mediante esta funcionalidad el usuario es capaz de ver en el mapa la información que contiene la matriz de demanda origen-destino, evitando el acceso a la base de datos donde se encuentra la información de la matriz y la búsqueda manual en el mapa. (Figura 2.5.3)
- *Interacción para la edición de recorridos.* Este tópico es fundamental para la manipulación de los casos de estudio. El mismo se maneja de tal forma que si bien restringe al usuario en cantidad de acciones disponibles, al mismo tiempo esto lo ayuda a evitar errores. Por ejemplo la edición sólo se puede realizar sobre el recorrido que se encuentre activo en el mapa y la misma debe ser finalizada, con el objetivo de verificar la correctitud del recorrido. (Figura 2.5.4)
- *Superposición de recorridos.* Este desafío fue principalmente visual, ya que el usuario debe ser capaz de distinguir varios recorridos que pasen sobre un mismo tramo. Para lo cual la herramienta despliega los recorridos a los lados de los tramos. (Figura 2.5.5)

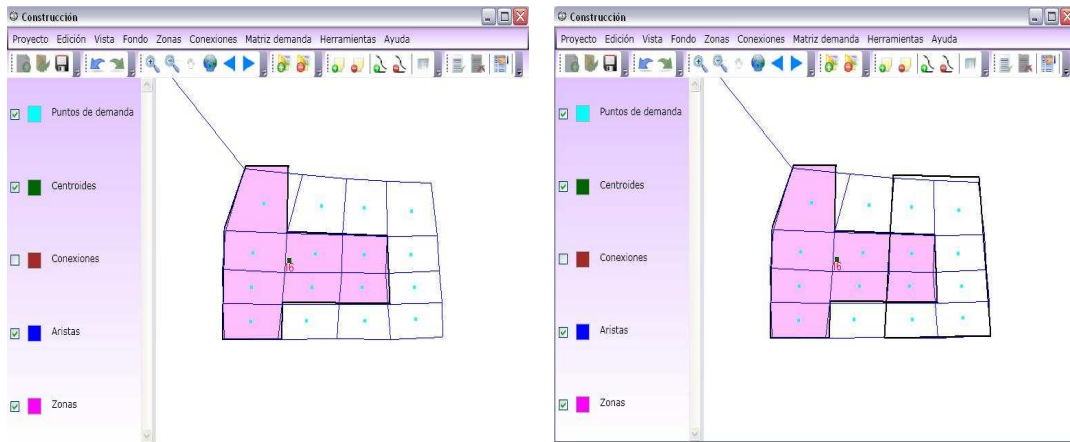


Figura 2.5.1

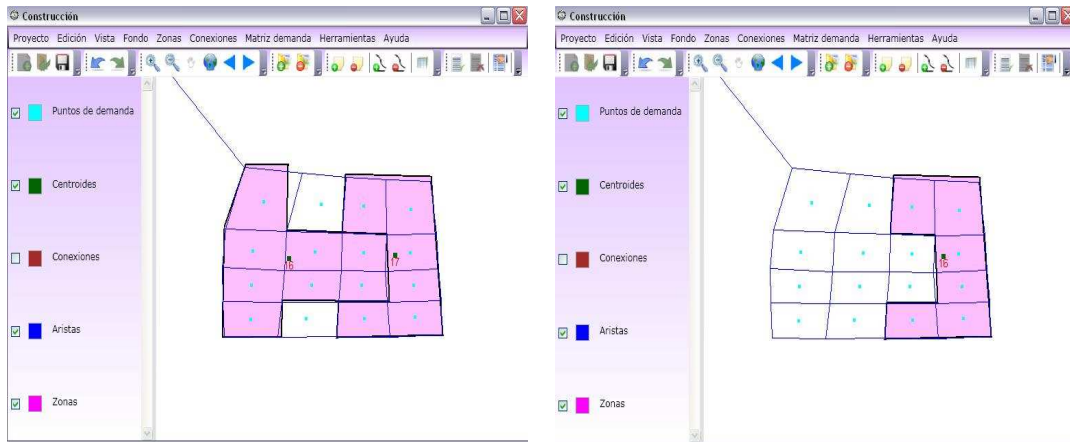


Figura 2.5.2

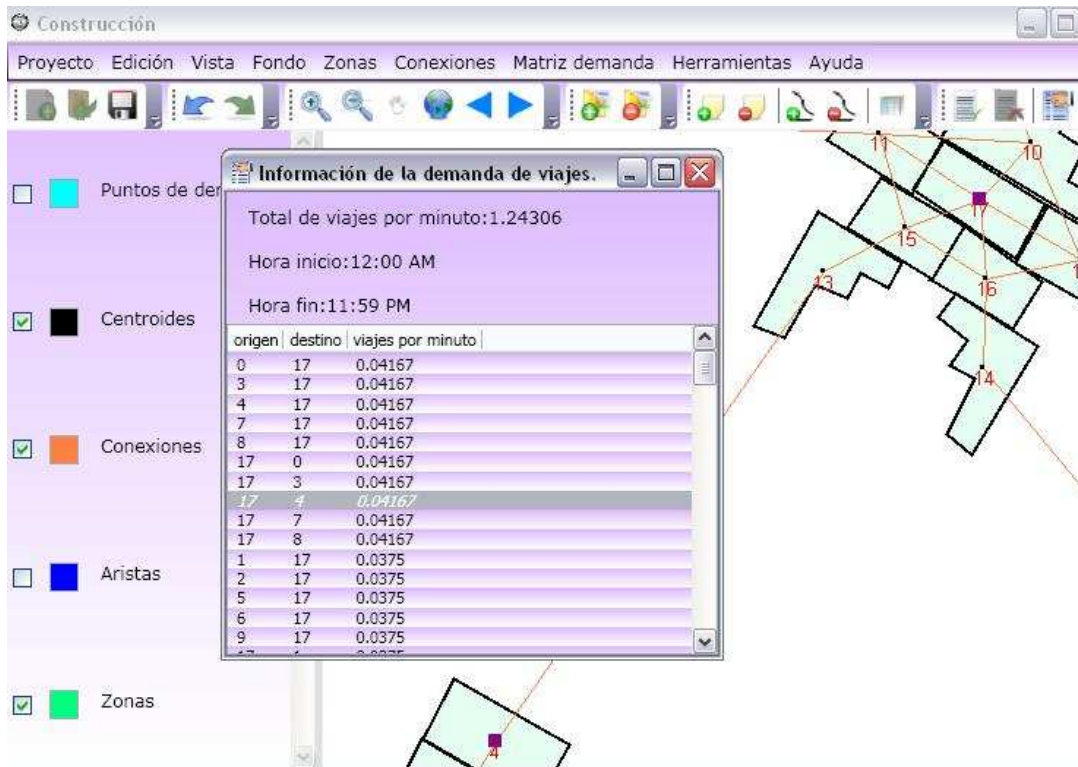


Figura 2.5.3

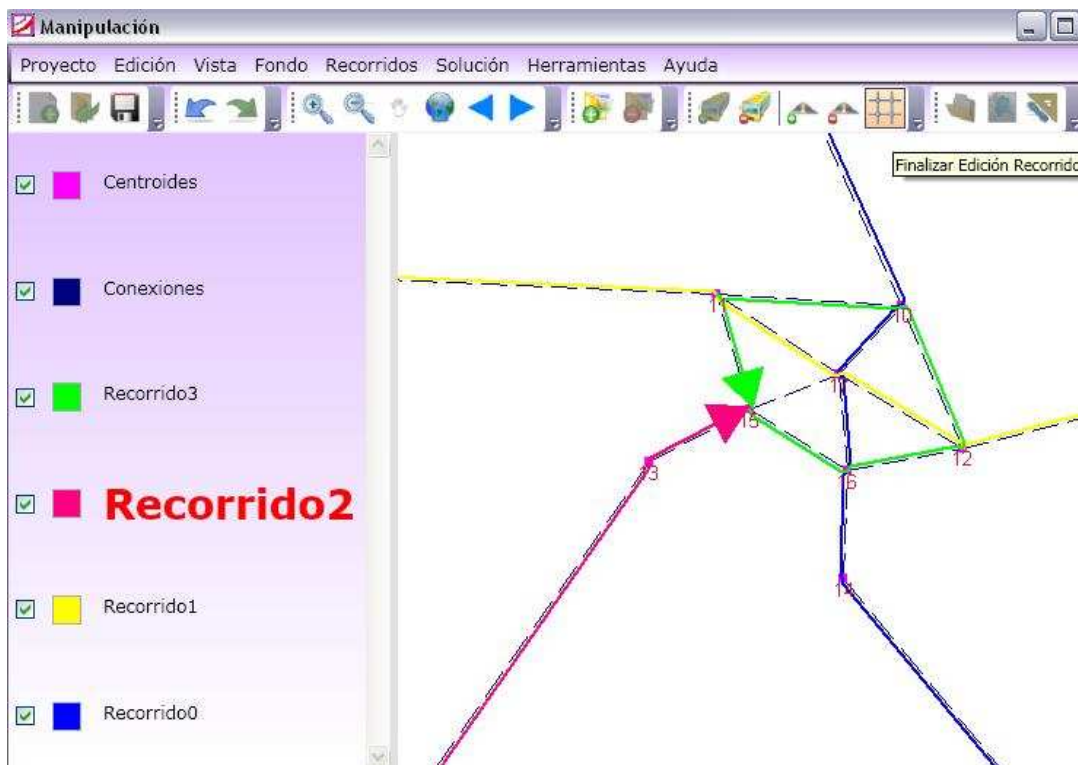


Figura 2.5.4

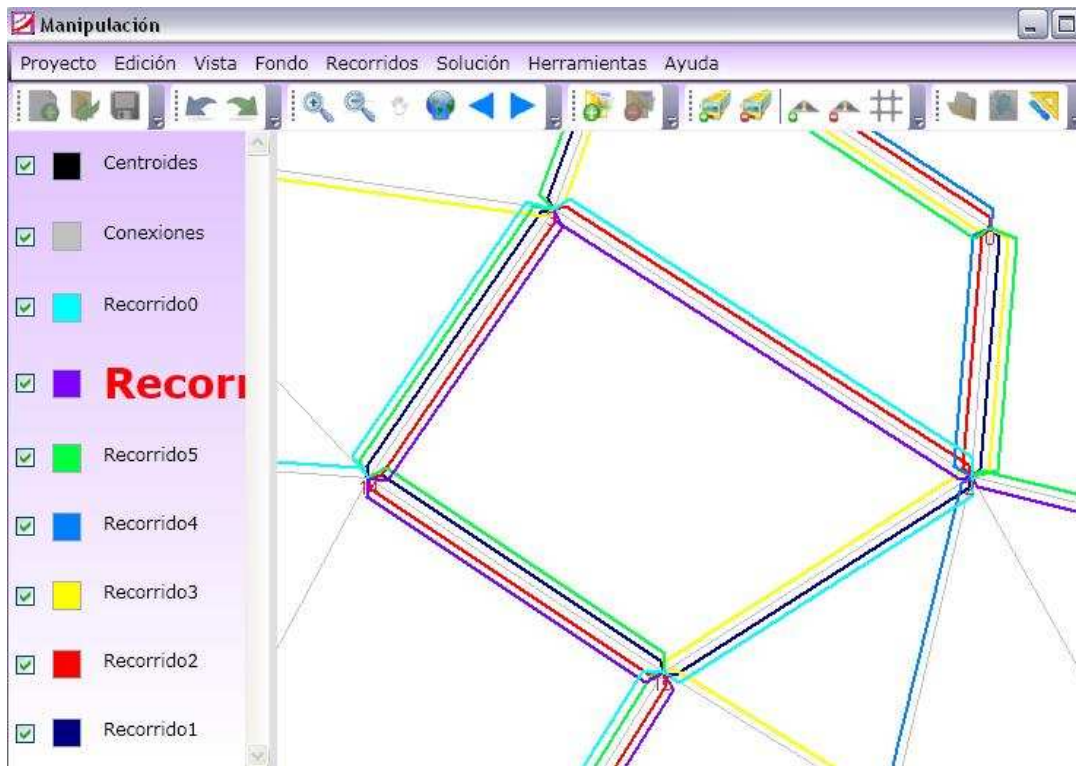


Figura 2.5.5

Al resolver los desafíos planteados igoR-tp debió superar algunas etapas de verificación que se detallan en el anexo correspondiente. (Anexo Verificación y Validación)

Una vez culminadas y aprobadas las pruebas realizadas, con la correspondiente validación del usuario, quien es el eje fundamental para el diseño de la herramienta, se resolvió liberar la primera versión completa de igoR-tp.

3. Conclusiones

Luego de atravesar todo el proceso de investigación, relevamiento de requisitos del usuario, diseño y arquitectura, estudio de la interacción entre la herramienta y el usuario, creemos que se ha cumplido con los objetivos planteados para este proyecto.

Hoy en día se cuenta con una herramienta estable con una interfaz gráfica integrada que permite el manejo de información geográfica para la optimización de la red de recorridos de transporte público.

Los resultados obtenidos están en concordancia con los esperados. La herramienta “igoR-tp” es un software que implementa las funcionalidades requeridas por el usuario, incluyendo la facilidad en su uso e instalación. Además los componentes externos que utiliza nuestro producto son de libre acceso.

Por otra parte se entrega un manual de usuario, el cual está accesible mediante el menú de Ayuda en “igoR-tp” así como en el CD de instalación, donde se describe entre otras cosas las funcionalidades que brinda “igoR-tp” y como se usan. Asimismo se agregó documentación técnica en el CD, para facilitar la continuación de nuestro trabajo para futuros emprendimientos.

Otro elemento que se entiende de gran relevancia es el informe que adjuntamos como anexo acerca del estudio de herramientas de tipo no propietarias para el manejo de información geográfica, en el mismo se especifican las ventajas y desventajas que se encontraron para poder emplearlas. (Anexo Búsqueda y Selección de Tecnologías SIG.)

También deseamos remarcar que se han preparado los datos del caso de estudio de Rivera para ser utilizados por los algoritmos de optimización, mediante los cuales se verificaron las funcionalidades que otorga “igoR-tp”. En el Anexo Caso de estudio: Ciudad de Rivera, se muestra parte de la creación de ese caso, junto a su experimentación, la cual fue realizada por el usuario.

3.1 Trabajos a Futuro

Si bien se ha cumplido con los objetivos planteados aún quedan muchos elementos por mejorar y explorar.

Al enfrentarse con herramientas nuevas no se contaba con la experiencia previa, para utilizarlas con la mayor parte de su potencialidad desde un principio, lo que trajo como consecuencia que varias ideas permanecieran en la teoría sin poder bajarlas a la práctica. Por ello ahora detallaremos cuales fueron para tenerlas presentes en futuros desarrollos.

- Mejorar algunas funcionalidades para brindarle mayor practicidad al usuario.

Para el módulo de construcción del caso de estudio, este hecho se refleja en las funcionalidades tanto de agregar zona como la de agregar conexiones. Por ejemplo podría existir una funcionalidad del tipo “generar zonas/conexiones automáticamente”, que mediante algún criterio, ya sea cantidad de manzanas, o rango de demanda, permitiera zonificar y crear las conexiones en una ciudad, esta funcionalidad facilitaría el trabajo del usuario sobre todo en ciudades de gran tamaño.

También sería positivo el manejo de atajos (shortcuts), puesto que mediante la combinación de ciertas teclas se podría acceder a algunas funcionalidades, un claro ejemplo sería que la funcionalidad de “Deshacer” estuviera ligada a la combinación CTRL+Z, como se acostumbra en la mayoría de aplicaciones donde se puedan editar elementos.

Otro aspecto a considerar sería diferenciar los distintos tipos de usuario que pueda tener la herramienta, para proporcionarle a cada uno de ellos lo que realmente necesita. En esta primera etapa iGoR-tp está enfocado al usuario diseñador de la red conocedor de los algoritmos de optimización empleados, por lo tanto capaz de manejar todas las funcionalidades. Sería interesante en una segunda etapa enfocarse también en el usuario planificador, que es experto en cuanto a la planificación de recorridos, pero no cuenta con la misma pericia para el manejo de este tipo de herramientas. Con lo cual sería de utilidad brindarle únicamente funcionalidades para visualizar recorridos, creando y evaluando soluciones. Por lo tanto una mejora para la herramienta implicaría en gestionar distintos tipos de usuario como ser diseñador de la red y planificador.

Por otra parte la herramienta podría manejar múltiples matrices de demanda origen destino en un mismo proyecto, de esta manera sería posible generar distintas soluciones para un caso de estudio teniendo ésta matriz como variable. Esta mejora se podría acompañar con el manejo de un conjunto de soluciones como una unidad, de esta manera se las podría comparar a un nivel más general.

De la misma forma si bien la visualización de recorridos superpuestos ha sido trabajada, esta se puede seguir mejorando, ya que si existen muchos recorridos que pasan sobre un mismo tramo y hay varios tramos a poca distancia la visibilidad pierde claridad. Una alternativa sería manejar líneas punteadas de distintos colores.

- Mejorar la comunicación entre el usuario y la herramienta.

Este punto es muy amplio y tiene varios frentes para enfrentarlo. Debido a ello vamos a ir desprendiendo elementos que servirían para apoyar la mejora sugerida.

La retroalimentación (feedback) indica en todo momento que es lo que está haciendo la aplicación, de esta manera ayuda al usuario a comprender mejor la forma en que trabaja. A pesar de que este tópico fue analizado intentando ser consistentes en toda la aplicación, mediante el uso de barras de progreso o deshabilitando de forma visible el acceso a las funcionalidades cuando era requerido, faltarían otros elementos que muestren el estado en que se encuentra la herramienta mediante alguna leyenda. Por ejemplo para el módulo de construcción del caso de estudio esa leyenda podría decir, “Proyecto abierto sin zonas”, “Proyecto abierto con zonas sin conexiones”, “Proyecto completo” (con zonas, conexiones y la matriz de demanda generada). También se podría indicar de alguna manera el nombre del proyecto abierto en algún sitio de la interfaz.

Por otra parte se intentó mantener un diseño uniforme a lo largo de toda la aplicación. Si bien se buscó que la herramienta tuviera una estructura similar a otras del mismo tipo (por ejemplo, ArcExplorer de ESRI), también se quiso que tuviera sus propias particularidades para diferenciarse del resto de los programas del sistema operativo, como ser el color del fondo de las ventanas de la aplicación. Siguiendo esta misma línea hay más elementos que permitirían independizarse aún más del sistema operativo sobre el que este ejecutando (Windows XP, Windows Vista), como son los contornos de las ventanas, el tipo de fuente utilizado para los títulos de las ventanas, las ventanas que permiten seleccionar algún color o las que se emplearon para explorar el sistema de archivos. Apoyándose en la tecnología usada para el desarrollo de igoR-tp, no es ilógico pensar en obtener esta independencia.

Para mejorar la comunicación con el usuario se pueden considerar otros elementos, por ejemplo que las entradas ingresadas en los respectivos campos estén alineadas hacia la izquierda o derecha de acuerdo a texto o número. También la calidad de los íconos se podría perfeccionar si se utilizarán herramientas más orientadas hacia ese fin, en nuestro caso los íconos fueron extraídos de distintos sitios en Internet y fueron mínimamente modificados. Inclusive se podría tomar asesoramiento de diseñadores considerando la posibilidad de independencia que brinda la herramienta WPF de Microsoft Visual Studio .NET, como se explica en el Anexo Entorno de desarrollo.

Otro ítem destacado para darle mayor fluidez a la comunicación que se ha ido describiendo es el diseño para el error. Actualmente igoR-tp cuenta con un manejo de errores que si bien notifica las fallas acontecidas, aún le falta trabajo para que esas notificaciones se conviertan en sugerencias, y que de esta forma el usuario tenga la información necesaria para poder resolver los errores con mayor facilidad.

4. Agradecimientos

Queremos agradecer a: Antonio Mauttone, María Urquhart, Gabriel Zuluaga, Miguel Gavilondo, y a todos aquellos que de una manera u otra colaboraron para que este proyecto saliera adelante.

Bibliografía

- [ABDLFPTVW 2008] C. Apa, J. Barreiro, S. De León, M. Freira, S. Pizard, J. Triñanes D. Vallespir, M. Wodzislawski. Curso Introducción a la Ingeniería de Software. 2008. Arquitectura de Software. -
<http://www.fing.edu.uy/inco/cursos/iis/wikiIIS/field.php/Curso/Curso>
(24/01/2009)
- [ABGGLLRS 2008] F. Alvarez, M. Belén, P. Garbussi, L. González, G. Llambias, G. López, R. Ruggia, R. Sosa. Curso de Taller de Sistemas de Información 1. 2008 .NET Framework -
<http://www.fing.edu.uy/inco/cursos/tsi/TSI1/2008/teorico/Introduccion.pdf>
(22/01/2009)
- [ABGGLLRS 2008] F. Alvarez, M. Belén, P. Garbussi, L. González, G. Llambias, G. López, R. Ruggia, R. Sosa. WPF Lifecycle, Layouts, Controles, DataBinding -
<http://www.fing.edu.uy/inco/cursos/tsi/TSI1/2008/teorico/clase4WPF.pdf>
(22/01/2009)
- [ABGGLLRS 2008] F. Alvarez, M. Belén, P. Garbussi, L. González, G. Llambias, G. López, R. Ruggia, R. Sosa. WPF – Aspectos generales -
<http://www.fing.edu.uy/inco/cursos/tsi/TSI1/2008/teorico/clase3WPF.pdf>
(22/01/2009)
- [AC 2008] Arquitectura 3 Capas -
<http://www.slideshare.net/Decimo/arquitectura-3-capas?nocache=8847>
(22/01/2009)
- [AM 2006] A. Américo, F. Martínez. Algoritmos Genéticos multi-objetivo aplicados al diseño de recorridos para transporte público. Proyecto de Grado de Ingeniería en Computación, Universidad de La República, 2006.
- [AS 2008] Material WPF - Capacitación Arkano Software.
www.arkanosoftware.com (22/01/2009)

- [ASP 2008] The ASP.NET Resource Index. Incogenius Inc. - <http://www.411asp.net/home/tutorial/howto/contentm/syndicat/rssstand> (22/01/2009)
- [AV 2006] A. Viklund. GeoTools. The Open Source Java GIS Toolkit. - <http://www.geotools.org/> (21/01/2009)
- [B 2009] B. Broscheit. FreeGIS. - <http://www.freegis.org/> (21/01/2009)
- [B 2008] D. Bartram. The Code Project – Your Development Resource - WPF Window Resizing. Enero 2008. http://www.codeproject.com/KB/WPF/WPF_Window_Resizing.aspx (22/01/2009)
- [BB 2007] Babylon On line Dictionary. 2007 - <http://www.babylon.com> (22/01/2009)
- [BBGG 1995] Bäcker, Buxton, Grudin, Greenberg. Human – Computer Interaction.
- [BEGMRPSP 2003] A. Bauzá, F. Escalas, F. Guaita Mas, M. Ruiz Pérez, J.M. Seguí Pons. La Planificación de Rutas de Transporte Escolar a través de un SIG: El Proyecto SIGTEBAL - Departamento de Ciencias de la Tierra, Laboratorio de Sistemas de Información Geográfica, Universidad de las Islas Baleares , España http://geofocus.rediris.es/docPDF/Articulo4_2003.pdf (18/12/2008)
- [BS 2008] E. Bocher , S. Steiniger. An Overview on Current Free and Open Source Desktop GIS Developments. Setiembre 2008.
- [CA 2009] Caliper Corporation. <http://www.caliper.com/> (13/03/2009)
- [CG 2008] Programas Standalone. Computer Glossary. - http://www.iwebtool.com/what_is_standalone.html (22/01/2009)
- [CI 2009] Citilabs. <http://www.citilabs.com/> (13/03/2009)
- [CM 2001] H. Caituiro Monge. Patrón Memento. Noviembre 2001. – <http://www.freewebz.com/amanecer/personal/papers/paper.memento.pdf> (22/01/2009)
- [CO 2008] COME S.A. <http://www.come.com.uy/i/02/> (03/02/2009)
- [COE 2008] Cooperativa de Obreros y Empleados del Transporte Colectivo. <http://www.coetc.com/index.htm> (03/02/2009)

- [CP 2008] The Code Project – Your Development Resource – Windows Forms
<http://www.codeproject.com/script/Forums/View.aspx?fid=387161&msg=2791219> (22/01/2009)
- [CU 2008] Compañía Uruguaya de Transportes Colectivos S.A.
<http://www.cutcsa.com/index.php?oid=9> (03/02/2009)
- [CWF 2007] Cómo: Crear controles de formularios Windows Forms.
Noviembre 2007. - <http://msdn.microsoft.com/es-es/library/bs3yhkh7.aspx>
(22/01/2009)
- [DH 2003] G. Desaulniers, M. D. Hickman. Public Transit. Reporte Técnico G-2003-77, GERAD, 2003.
- [DN 1986] Draper, Norman. User Centered System Design. Enero 1986.
- [EA 2008] Diccionario de Infomática. Exception – Alegsa. -
<http://www.alegsa.com.ar/Diccionario/diccionario.php> (21/01/2009)
- [EGG 2008] Essays on Geography and GIS. Setiembre 2008 -
<http://www.esri.com/library/bestpractices/essays-on-geography-gis.pdf>
- [FDO 2007] FDO Data Access Technology - <http://fdo.osgeo.org/> (21/01/2009)
- [FL 2007] E. Fernández, T. Lorenzo. Material del curso Interacción persona computadora. Cecal – Inco – Fing – UdelAR.
www.fing.edu.uy/inco/cursos/inpercom (18/12/2008)
- [FSF 2008] Free Software Foundation - <http://www.fsf.org/> (21/01/2009)
- [G 2005] I. N. Gregory. A place in history: A guide to using GIS in historical research. 2nd Edition. Centre for Data Digitisation and Analysis, Queens University, Belfast, Belfast, Northern Ireland, BT7 1NN
- [GCM 2005] G. Garula, D. Cukerman, M. Melgar Herramienta para el diseño de transporte de recorridos en transporte colectivo urbano. Proyecto de Grado de Ingeniería en Computación, Universidad de La República, 2005.
- [GG 2009] GRASS GIS - <http://grass.osgeo.org/index.php> (21/01/2009)
- [GIS 2008]GIS.com - The Guide to Geographic Information Systems -
www.gis.com (22/01/2009)
- [GL 2005] Tipos de Licencia de Software. R. M. Gómez Labrador. Septiembre 2005. -<http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf>
(21/01/2009)

- [GNU 2008] GNU Operating System - <http://www.gnu.org/home.es.html> (21/01/2009)
- [GT 2009] GeoTools InfoSheet. - <http://www.osgeo.org/geotools> (21/01/2009)
- [GV 2008] gvSIG. Conselleria de Infraestructuras y Transporte.: <https://gvsig.org/web/> (01/07/2008), <http://www.gvsig.gva.es/index.php?id=gvsig&L=0> (21/01/2009)
- [HCF 2007] Tutorial: Hospedar un control de formularios Windows Forms en Windows Presentation Foundation mediante el uso de XAML. Noviembre 2007. - <http://msdn.microsoft.com/es-es/library/ms742875.aspx> (22/01/2009)
- [HWF 2007] Cómo: Heredar de controles de formularios Windows Forms existentes - <http://msdn.microsoft.com/es-es/library/7h62478z.aspx> (22/01/2009)
- [JP 2003] The JUMP Project. - <http://www.jump-project.org/> (21/01/2009)
- [K 2007] P. Kinlan. C# .NET Framework Blog. 2007 - <http://www.kinlan.co.uk/2005/06/xaml-menu-example.html> (22/01/2009)
- [LG 2004] F. A. Lara González. VISUALIZADOR DE DATOS GEOGRÁFICOS SOBRE INTERNET. 2004. - <http://biblioteca.uct.cl/tesis/freddy-lara/tesis.pdf> (22/01/2009)
- [M 2008] C. Mayr. Material del curso de Administración y Seguridad de Sistemas
- [M 2005] A. Mauttone. Optimización de Recorridos y Frecuencias en Sistemas de Transporte Público Urbano Colectivo. Tesis de Maestría en Informática PEDECIBA, Universidad de La República, 2005.
- [M 2002] E. Manchón. ¿Qué es la usabilidad? - [ISO/IEC9126] - [ISO/IEC9241]. Barcelona. Enero 2002. - http://www.ainda.info/que_es_usabilidad.htm (22/01/2009)
- [M 2001] G. A. Miller. The Magical Number Seven Plus or Minus Two, Some Limits on Our Capacity for Processing Information. 2001.
- [M 1995] A. Marcus. Principles of effective visual communication for graphical user interface design. 1995
- [MCT 2008] Menu ControlTemplate Example - <http://msdn.microsoft.com/en-us/library/ms752296.aspx> (22/01/2009)

- [MG 2007] L. Muro García. Licencias de Software. Facultad de Ciencias. Universidad Nueva Esparta. Caracas, Venezuela. Noviembre 2007 - <http://www.monografias.com/trabajos55/licencias-de-software/licencias-de-software.shtml> (21/01/2009)
- [MF 2003] A. Meneghin, A. Ferraz-Leite. Interfaz Gráfica para Herramienta de Planificación de Transporte Público Urbano Colectivo. Proyecto de Grado de Ingeniería en Computación, Universidad de La República, 2004.
- [MI 2009] Microsoft COM: Component Object Model Technologies. 2009 <http://www.microsoft.com/com/default.mspix> (12/03/2009)
- [MM 2004] Definición y significado de Amigable. MasterMagazine. 2004. - <http://www.mastermagazine.info/termino/3834.php> (22/01/2009)
- [MP 2007] The Memento Pattern. CMSC 433. Abril 2007. - <http://www.cs.umd.edu/~atif/Teaching/Spring2007/Lectures/MementoPattern.pdf> (22/01/2009)
- [MS 2008] How to Insert element TextBox or ComboBox in xaml file. Microsoft Silverlight. 2008 - <http://silverlight.net/forums/p/1688/4175.aspx> (22/01/2009)
- [MS 2001] H. J. Miller y S-L. Shaw. Geographic Information Systems for Transportation.
- [MS 2000] H. J. Miller y S-L. Shaw. GIS-T DATA MODELS. http://www.gisvisionmag.com/Book/miller_shaw.pdf (21/01/2009)
- [MW 2008] Map Windows GIS. Julio 2008. - <http://www.mapwindow.org/> (21/01/2009)
- [N 2006] T. Neward. Arquitectura pragmática Disposición en capas. Noviembre 2006 - http://www.microsoft.com/spanish/msdn/articulos/archivo/271106/voices/NPAL_ayering.mspix (22/01/2009)
- [N 2004] Understanding Metadata. NISO Press. National Information Standards Organization 2004. ISBN 1-880124-62-9
- [N 1990] D. A. Norman. The Design of Everyday Things. Febrero 1990.
- [NW 2006] T. Northrup, S. Wildermuth. Microsoft .NET 2.0 Application Development Foundation, Training Kit. Julio 2006.

- [OGC 1994] OGC Open Geospatial Consortium - <http://www.opengeospatial.org/> (21/01/2009)
- [OK 1996] I. H. Osman, J. P. Kelly. Meta-heuristic: Theory and Applications. Springer 1996. ISBN 0792397002, 9780792397007
- [ORW 1994] Odoni, Rousseau, Wilson .Urban Transit.
- [OS 2008] OpenSourceGIS. Diciembre 2008. - <http://opensourcegis.org/> (21/01/2009)
- [OSI 2007] Open Source Initiative - <http://www.opensource.org/> (21/01/2009)
- [OSG 2008] OSGeo The Open Source Geospatial Foundation - <http://www.osgeo.org/> (21/01/2009)
- [OW 1990] J. de D. Ortúzar, L. G. Willumsen. Modelling Transport Second Edition. 1994
- [PC 2008] PCMAG.COM. The independent guide to technology. 2008 http://www.pcmag.com/encyclopedia_term/0,2542,t=NFS&i=47963,00.asp# (12/03/2009)
- [PFC 2008] Definición GPU. Proyecto fin de carrera.com - <http://www.proyectosfindecarrera.com/definicion/gpu.htm> (22/01/2009)
- [R 2000] J. Raskin. Human Interface. Publicado en el año 2000.
- [RAE 2009] Diccionario de la Real Academia Española. Versión 22ª. <http://www.rae.es/rae.html> (12/03/2009)
- [SL 2005] Software Licencias. Marzo 2005 - http://www.proint.info/wiki/index.php?title=Software:_Licencias (21/01/2009)
- [S 2009] S. Stefan. An Overview of Free & Open Source Desktop GIS. Enero 2009. <http://www.spatialserver.net/osgis/> (21/01/2009)
- [TU 2003] El Transporte Urbano. Enero 2003. http://www.geocities.com/capecanaveral/hall/6994/TUM_en_el_mundo.htm (03/02/2009)
- [UC 2008] Unión Cooperativa Obrera del Transporte. <http://www.ucot.net/> (03/02/2009)
- [VCP 2007] Variedades de controles personalizados. Noviembre 2007. - <http://msdn.microsoft.com/es-es/library/ms171725.aspx> (22/01/2009)

- [VG 2007] Juan Ignacio Varela García. Estudio de un proyecto de software libre cuyos creadores originales abandonan el desarrollo. Caso de estudio: JUMP GIS -http://74.125.113.132/search?q=cache:jno40dotGOoJ:https://svn.forge.morfeo-project.org/svn/freeswmaster/tags/20072008/trunk/nacho/Work1/nachovarela_work.pdf+proyecto+JUMP&hl=es&ct=clnk&cd=4&gl=uy&client=firefox-a (21/01/2009)
- [Z 1999] M. Zeiler. Modeling our World, The ESRI Guide to Geodatabase Design. ESRI Press 1999. ISBN 1-879102-62-5

5. Anexo. Caso de estudio: Ciudad de Rivera

Recordando los objetivos del PDT (1.1 Contexto del proyecto), esta sección describe parte de la tarea llevada a cabo (por el PDT) para cumplir con el objetivo 4 - obtención de datos de calidad. También describe la utilización de la herramienta iGoR-tp por parte del usuario diseñador de la red para la construcción de una propuesta relativa al objetivo 6 - solución alternativa de recorridos para la ciudad de Rivera.

5.1 Objetivo 4 - Obtención de datos de calidad

La obtención de datos de calidad es un proceso muy costoso. Los datos a los que se hace referencia corresponden a información de demanda y a información geográfica. En esta sección se describen dos tareas importantes en el PDT con respecto a la obtención de la información mencionada. El procesamiento de una encuesta, de la cual se obtuvo la información de demanda, y la digitalización de capas de la ciudad, de la cual se obtuvo la información geográfica. Un integrante del grupo del Proyecto de Grado, Manuel Martínez, participó en las actividades descritas en las siguientes secciones.

5.1.1 Procesamiento de la encuesta de hogares

La información de demanda se obtuvo a partir de una encuesta de hogares realizada en el año 2007 para relevar información de movilidad de la ciudad de Rivera.

El procesamiento de la encuesta de hogares tiene como objetivo la construcción de una base de datos con la información recogida en la encuesta. Esto se llevó a cabo en diferentes etapas: (1) pasaje de los formularios a planilla electrónica, (2) normalización de criterios y unificación de categorías de la información, (3) cálculo de orígenes y destinos, (4) construcción de una base de datos.

(1) Para digitalizar la información de los formularios se utilizó el programa Microsoft Excel. El formato de la planilla es una columna por campo en el formulario. El pasaje y la posterior verificación de los datos desde los formularios a la planilla electrónica es una tarea costosa debido al tiempo que requiere.

(2) La etapa de normalización de criterios se llevó a cabo principalmente debido a que los encuestadores no siempre se restringieron a los posibles valores de los campos

del formulario (indicados en el mismo). Durante el pasaje de datos se trató de incluir toda la información que ingresaron los encuestadores, esto agregó nuevos valores que debieron ser normalizados. Luego de tener la información normalizada se debió realizar una unificación de valores de manera que estos tuvieran peso estadístico.

(3) Uno de los propósitos del proyecto PDT era construir una matriz origen destino de los viajes en ómnibus a nivel de número de manzanas. Esto quiere decir que los orígenes y destinos de los viajes debían ser identificados por números de manzana. Para esto se debió hacer corresponder cada valor de origen destino de los viajes en bus con su correspondiente número de manzana. Este trabajo no resultó sencillo dado que los valores ingresados por los encuestadores en los mencionados campos no siempre representaban un punto geográfico identificable. Para estos casos se utilizaron los datos, duración del viaje, línea de bus y lugar geográfico del hogar (siempre que se contara con estos).

(4) Una vez que se tuvieron los datos de la encuesta en una planilla electrónica se decidió mantener la información también en una base de datos, para poder manipularlos de forma más eficiente. Se decidió utilizar Microsoft Access dado que es un buen manejador para pequeñas y medianas base de datos.

5.1.2 Digitalización de capas geográficas de la ciudad

o Capa de manzanas y centroides

Dado que la información de los viajes en ómnibus de la encuesta en la base de datos habían sido llevados al nivel de número de manzana, se debía hacer corresponder estos números con alguna representación geográfica. Se decidió utilizar como representación de las manzanas, puntos. El punto representa el centroide de la manzana. Para esto por supuesto era necesaria una capa de polígonos que representen las manzanas de la ciudad. Se consiguió a través del Instituto de Agrimensura una capa de manzanas de la ciudad. Esta capa tenía unos años de creada y no había tenido actualizaciones. El primer paso entonces fue actualizar la capa de manzanas. Primero se actualizaron los números de manzanas de la capa dado que no se encontraban todos correctos, luego se agregaron todas las manzanas de donde partía o finalizaba un viaje y que no se encontraba representada en la capa. Una vez que se contó con una capa de

manzanas actualizada para la tarea se llevó a cabo un procedimiento para crear la capa de centroides a partir de la capa de manzanas.

○ **Capa de ejes de calles**

De la misma manera que la capa de manzanas, la capa de ejes también se consiguió a través del Instituto de Agrimensura. Esta capa también se encontraba desactualizada. Para completar la capa de ejes de calles se usó un archivo de AutoCAD que posee la Intendencia de Rivera el cual contiene la red de calles de la ciudad actualizada. La capa además necesitó ciertos ajustes para los cuales fue necesario volver a digitalizar muchas de las calles. Por último era importante identificar las calles por esto hubo que completar los nombres de las calles de la capa ya que la mayoría no contaba con éste.

○ **Capa de paradas**

En el viaje realizado a la ciudad de Rivera en el mes de junio del 2007, se realizó un registro de las posiciones de las paradas de esta ciudad. Para esta tarea se utilizó un GPS y navegador de mano, el cual registraba las paradas con un error en el entorno de los 10 metros, dependiendo de la interferencia creada por las edificaciones en el punto donde se encontraba la parada. Además del GPS se registró en un mapa de papel de la ciudad, las posiciones de las paradas, utilizando un marcador. También se registraron los valores X e Y brindados por el GPS para tener un respaldo. Las paradas son identificadas por el orden en que se tomaron las medidas. Se tomaron un total de 244 medidas que representaban 244 paradas. Para hacer más ágil el registro de las paradas, si se encontraba una parada enfrente a otra o esquina cruzada, se registraba una parada con el GPS y la otra se la registraba sólo en el mapa con el mismo identificador que la medida por el GPS, pero agregándole prima en el caso que fuesen dos o, prima y dos prima en el caso en que fuesen tres.

Luego de tener las medidas se construyó la capa de paradas con las medidas tomadas con el GPS. Esta capa tiene proyección UTM marco estándar de GPS, sistema de referencia WGS84 y sistema de proyección UTM zona 21 sur. La capa de calles y la capa de manzanas que se utilizan se encuentran en la proyección de la cartografía oficial, ROU-USAMS o Gauss. Debido a esto se re proyectó la capa de paradas a la proyección de la de calles y manzanas.

Por lo mencionado, a esta capa se le agregó las paradas prima y dos primas que no habían sido registradas con el GPS. Además de corregir las posiciones de algunas

medidas que, ya fuese por el error generado por el GPS o por el error de las capas de calles y de manzanas, se encontraban del lado incorrecto de la calle, sobre las manzanas, o en lugares donde no llegaban las calles.

- **Capa de líneas de ómnibus**

Una vez que se contó con la capa de ejes de calle de la ciudad se comenzó con la digitalización de las líneas de ómnibus actuales de la ciudad. A partir de la información brindada por la Intendencia de Rivera acerca del trayecto de los recorridos, se digitalizaron las mencionadas líneas.

Se digitalizaron además, las líneas de ómnibus de una propuesta realizada a la Intendencia de la ciudad en el año 2007 por parte de una consultoría.

En ambos casos se creó una capa para la ida de una línea y una capa para la vuelta.

5.2 Objetivo 6 – Construcción de una propuesta alternativa

En esta sección se mostrarán algunos resultados de la utilización de la herramienta igoR-tp, por parte del usuario diseñador de red Antonio Mauttone, además tutor de este proyecto, tanto en el armado del caso de estudio como en algunos experimentos utilizando los algoritmos construidos en el departamento.

5.2.1 Armado del caso de estudio

Para el armado del caso de estudio se utilizaron los datos obtenidos por el PDT. Específicamente se utilizaron de la información geográfica: las capas de ejes de calles como aristas y las capas de centroides de manzanas como puntos de demanda. Se construyó una base de datos especial con el formato de tabla que utiliza la herramienta a partir de los datos de la base construida por el PDT.

A continuación se muestran una serie de figuras que describen distintos pasos llevados a cabo por el usuario para el armado del caso.

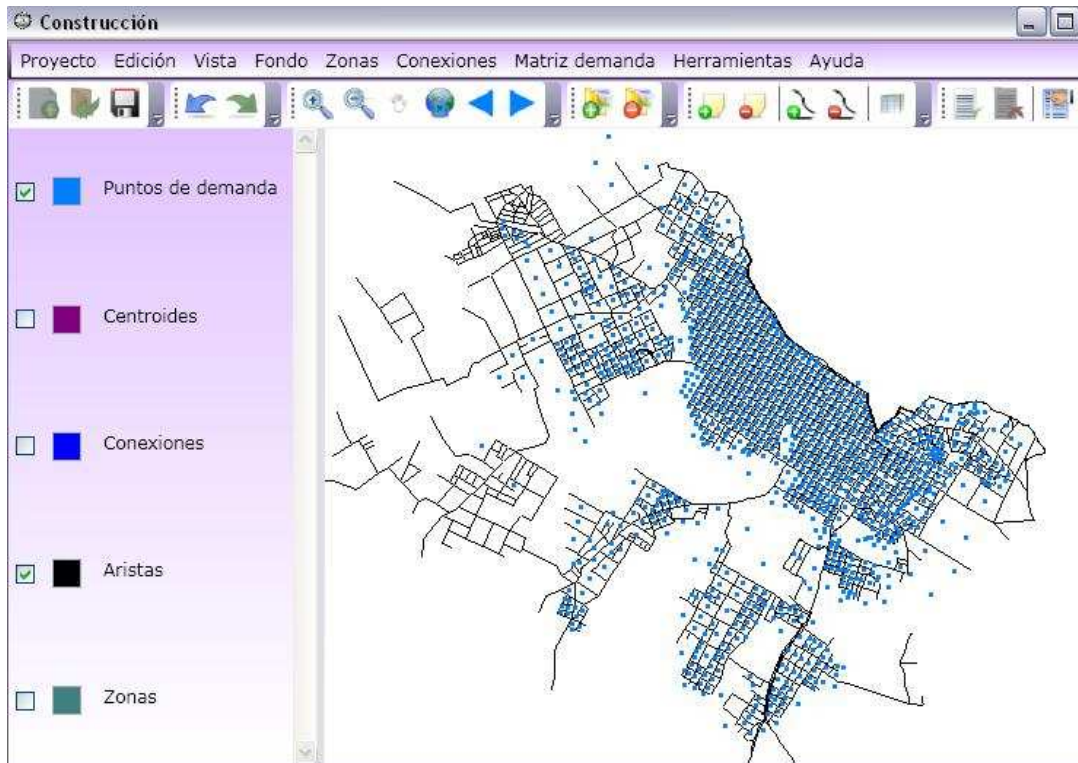


Figura 5.2.1.1. Esta captura muestra los datos de entrada, en negro la capa de aristas, en azul claro los puntos de demanda. El usuario se encontraba al comienzo de la tarea del armado.

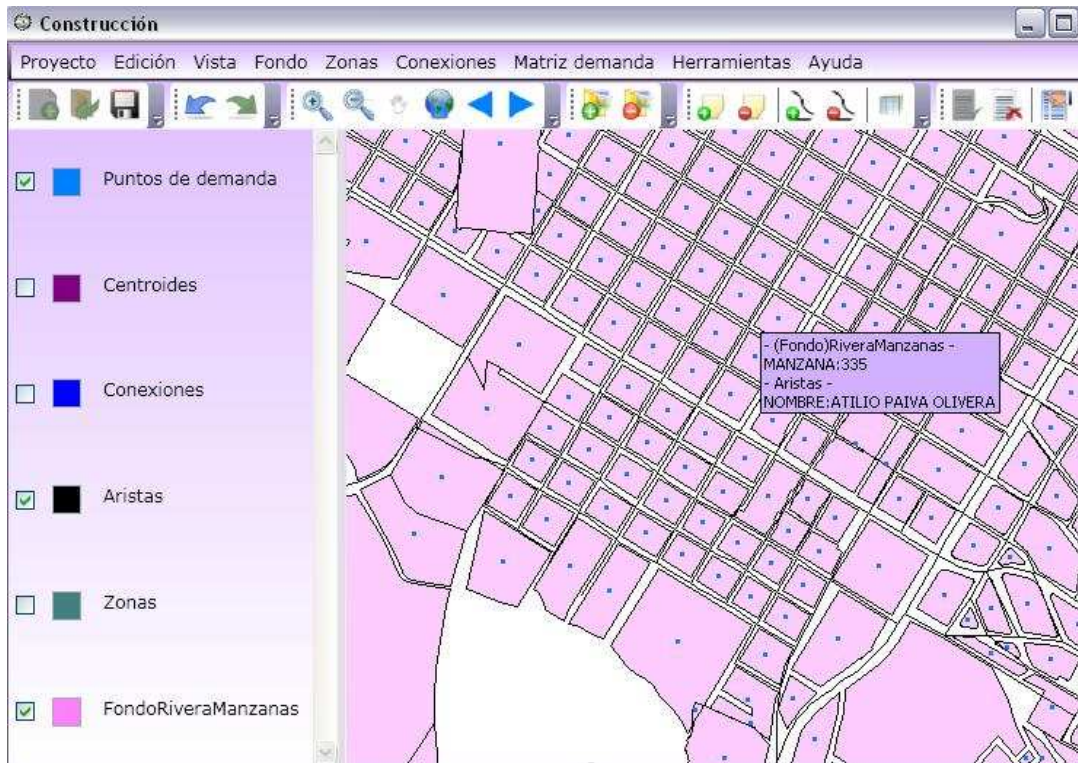


Figura 5.2.1.2. En esta captura se aprecia al usuario utilizando la herramienta “notas del mapa”. Esta herramienta es útil cuando se desea visualizar algunos atributos de las capas del mapa en un lugar específico. Las notas aparecen cuando el puntero del ratón se detiene sobre el mapa y desaparece al moverlo nuevamente. Los atributos que se visualizan son configurables. En la captura el usuario necesitaba saber el nombre de la calle de la capa de “Aristas” y el número de las manzanas de la capa de fondo “RiveraManzanas”.

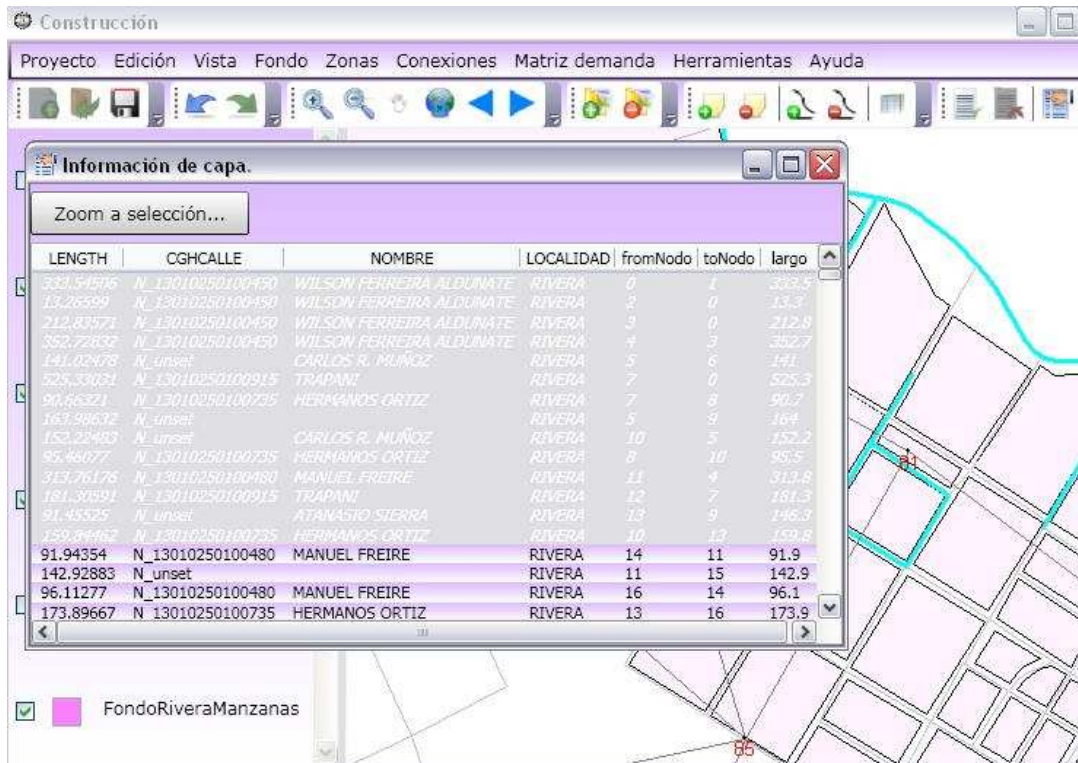


Figura 5.2.1.3. En esta captura se aprecia al usuario utilizando la herramienta “información de capa”. Esta herramienta es útil cuando se desea ver todos los atributos de una de las capas del mapa. Además permite realizar zoom a los elementos del mapa a los cuales pertenecen los atributos de las filas, cada fila corresponde a los atributos de uno de los elementos de la capa. Se puede apreciar además que la selección es marcada en el mapa para diferenciar los elementos seleccionados del resto de elementos de la capa.

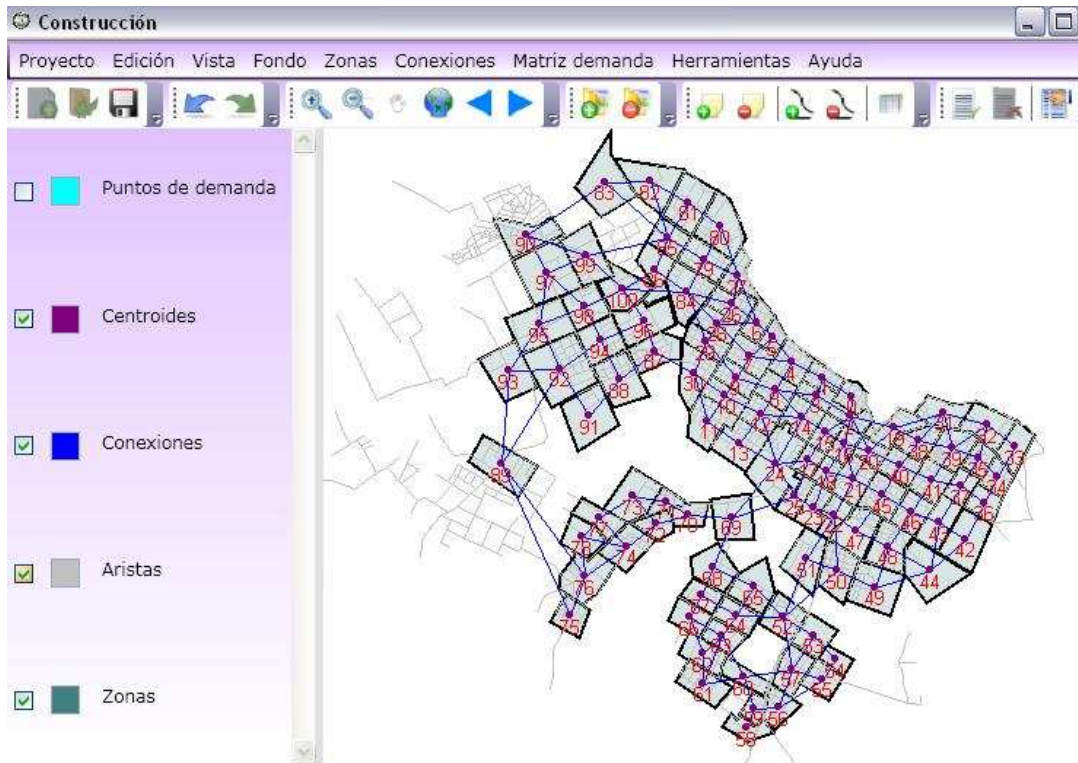


Figura 5.2.1.4. Esta captura muestra las capas de zonas, conexiones y centroides, luego de zonificar la ciudad y de conectar las zonas. Las zonas se ven en gris, los centroides en violeta y las conexiones entre zonas en azul.

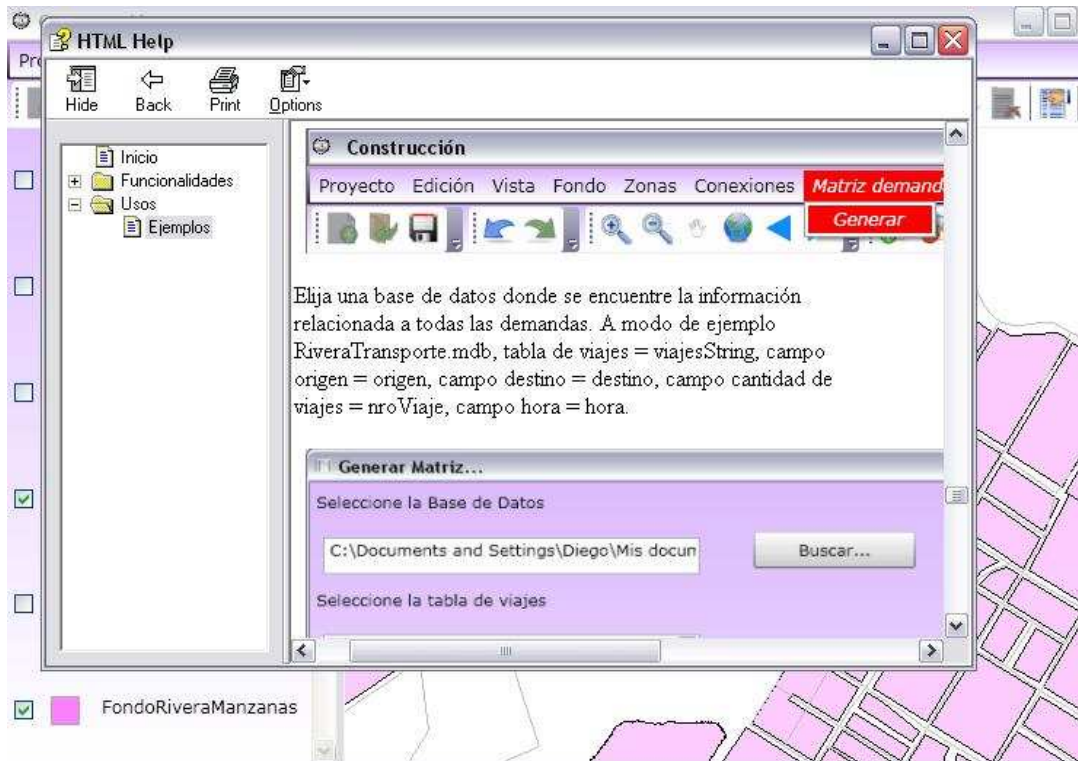


Figura 5.2.1.5. Esta captura muestra la ayuda del módulo de construcción. Se puede ver que el usuario está por generar la matriz demanda. La ayuda contiene una explicación de las funcionalidades y un ejemplo de las mismas.

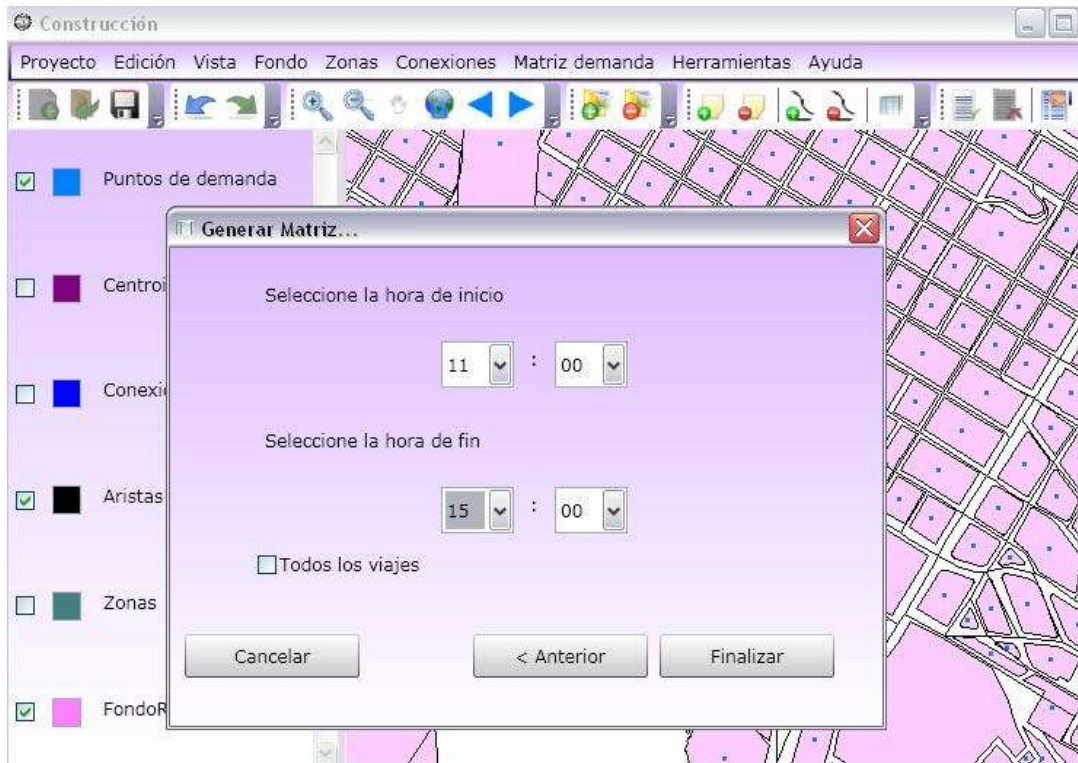


Figura 5.2.1.6. En esta captura se muestra al usuario en el proceso de creación de una matriz demanda, específicamente está seleccionando el rango horario.

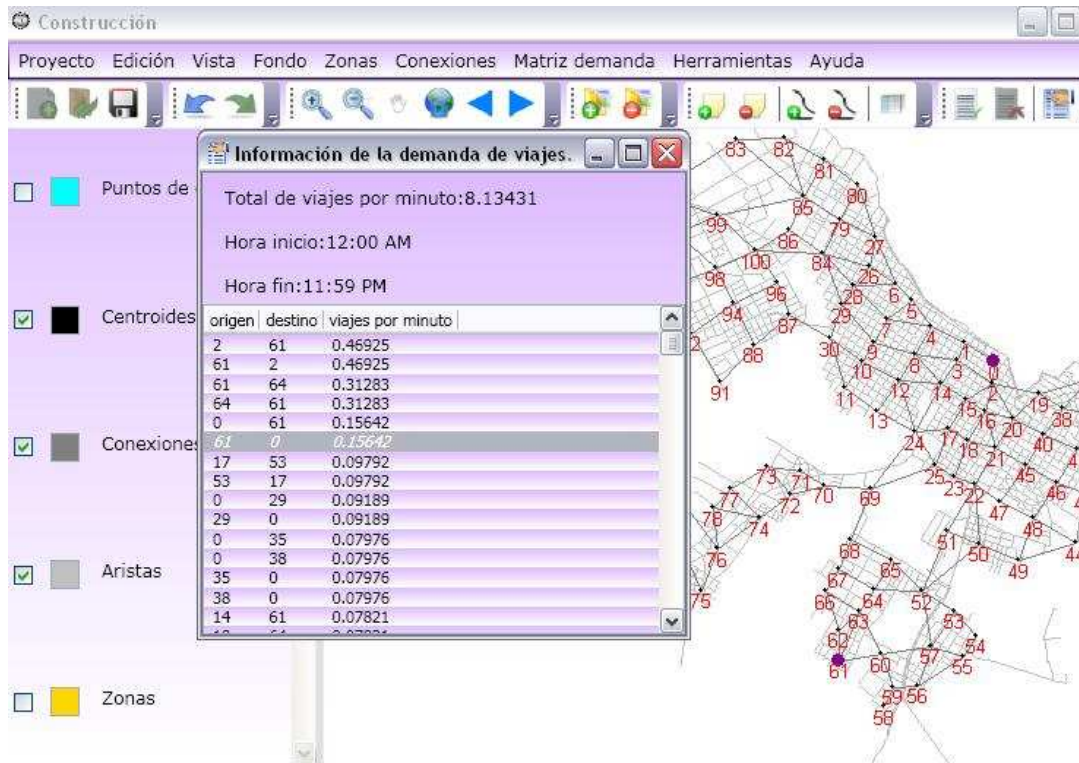


Figura 5.2.1.7. Esta captura muestra al usuario utilizando la herramienta “datos de demanda”. Esta herramienta es útil para observar los resultados de la matriz de demanda construida. Se puede ver que la tabla despliega la demanda entre las zonas, ordenada de mayor a menor. Además al seleccionar una fila en la tabla se visualizan en el mapa los centroides correspondientes a esa demanda, marcados en violeta.

5.2.2 Experimentos con el caso de estudio

Aquí se muestran capturas de los resultados de la generación de una solución utilizando los algoritmos y su posterior evaluación. La entrada para este proceso son las capas de centroides y conexiones generadas en el armado del caso.

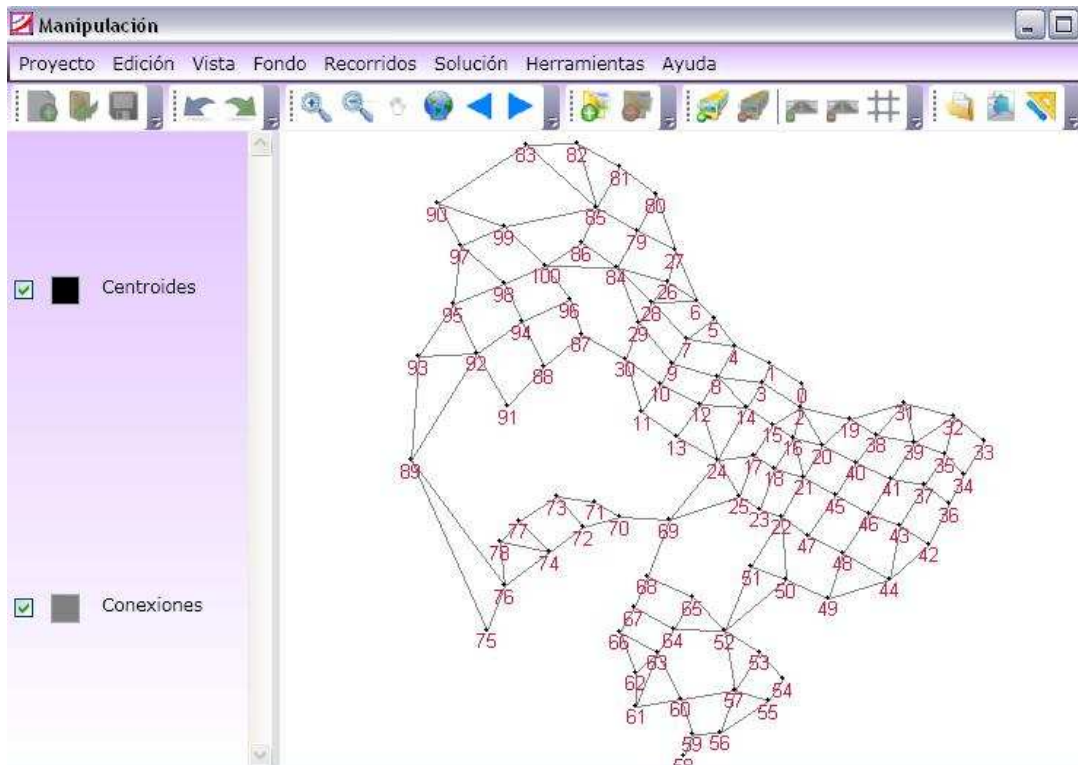


Figura 5.2.2.1. En esta captura se pueden apreciar las dos capas de entrada. En negro la capa de centroides y en gris la capa de conexiones.

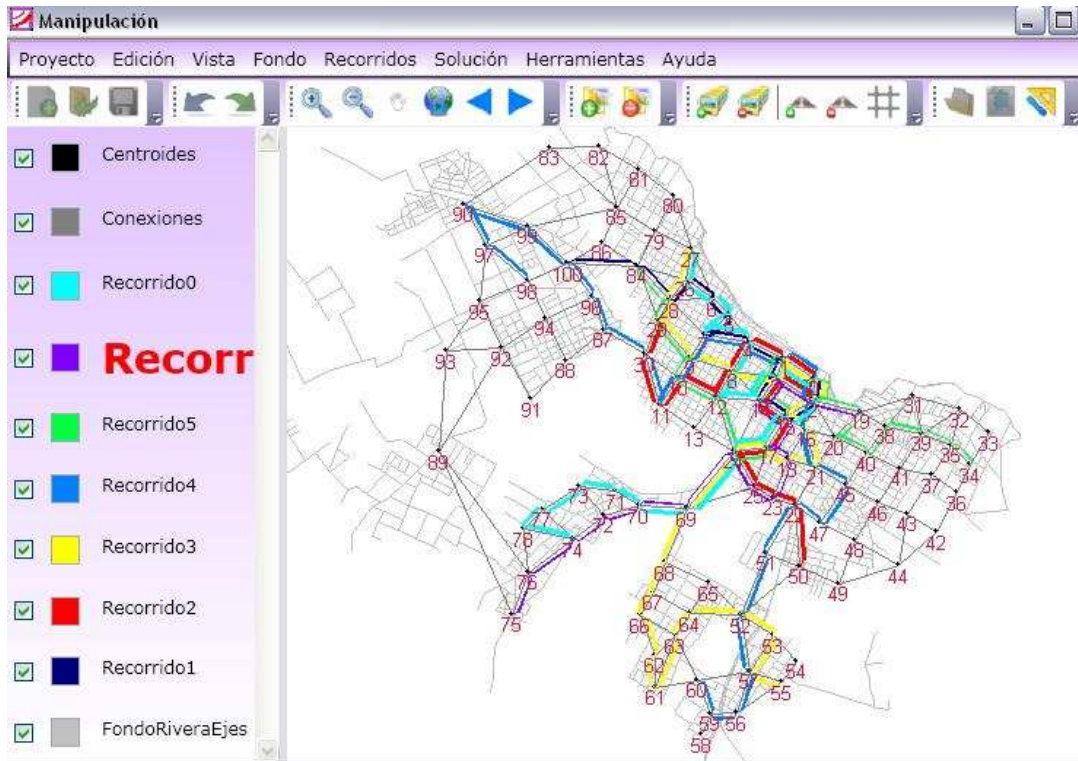


Figura 5.2.2.2. Esta captura muestra una solución generada por los algoritmos de optimización. La solución cuenta con 7 recorridos. Se puede observar que los distintos recorridos no se superponen y que además no tienen el mismo grosor de línea, esto último se debe a que el usuario se encuentra en la vista de recorridos “ponderada por frecuencia”, a mayor frecuencia mayor ancho de línea.

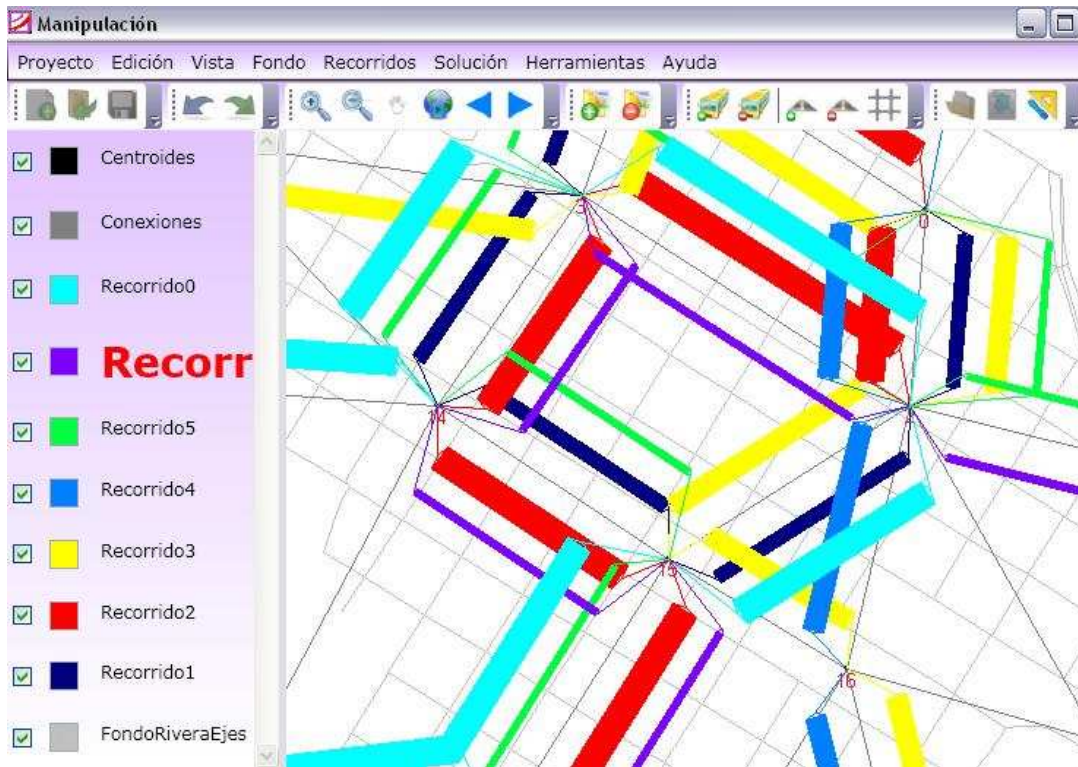


Figura 5.2.2.3. Esta captura muestra la ciudad y los recorridos de la solución a un mayor nivel de acercamiento. Se muestra de manera más clara la diferencia entre las frecuencias de los recorridos como consecuencia del ancho de estos. La visión de cerca de la solución con vista ponderada en una zona donde pasan muchos recorridos puede resultar confusa.

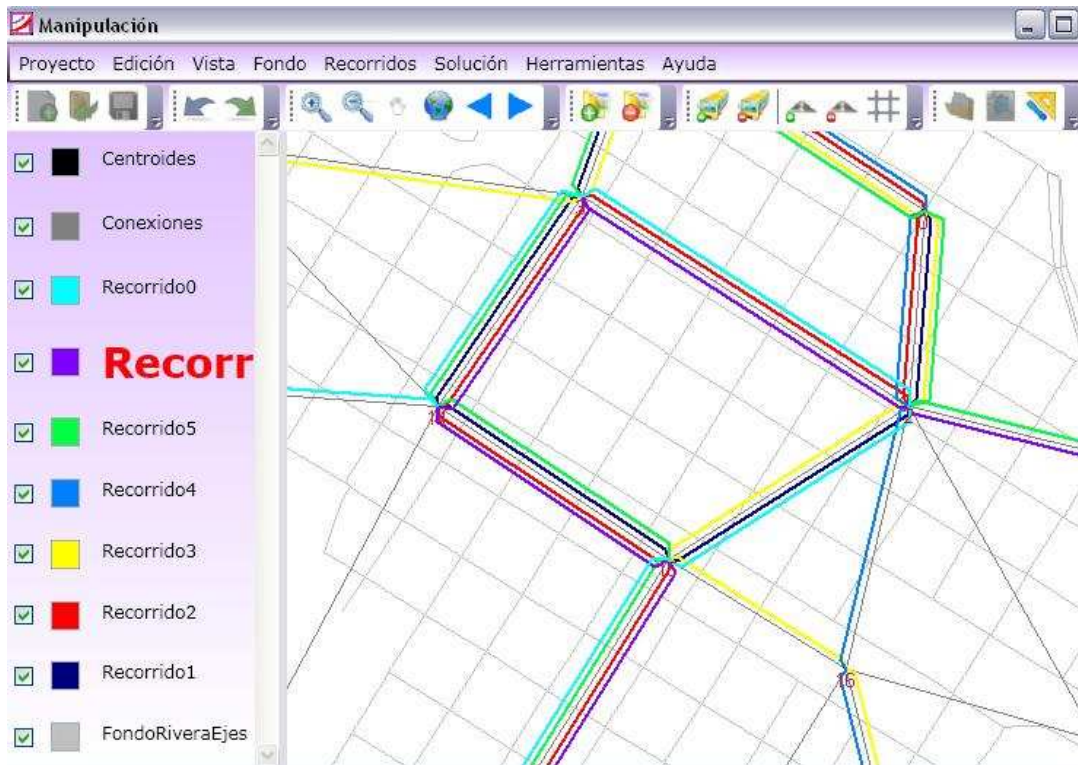


Figura 5.2.2.4. Esta captura tiene la misma escala que la imagen anterior pero utilizando la vista de recorridos “normal”. El usuario puede configurar las vistas y algunos de sus parámetros. La visión “normal” resulta menos confusa de cerca que la visión “ponderada por frecuencia”.

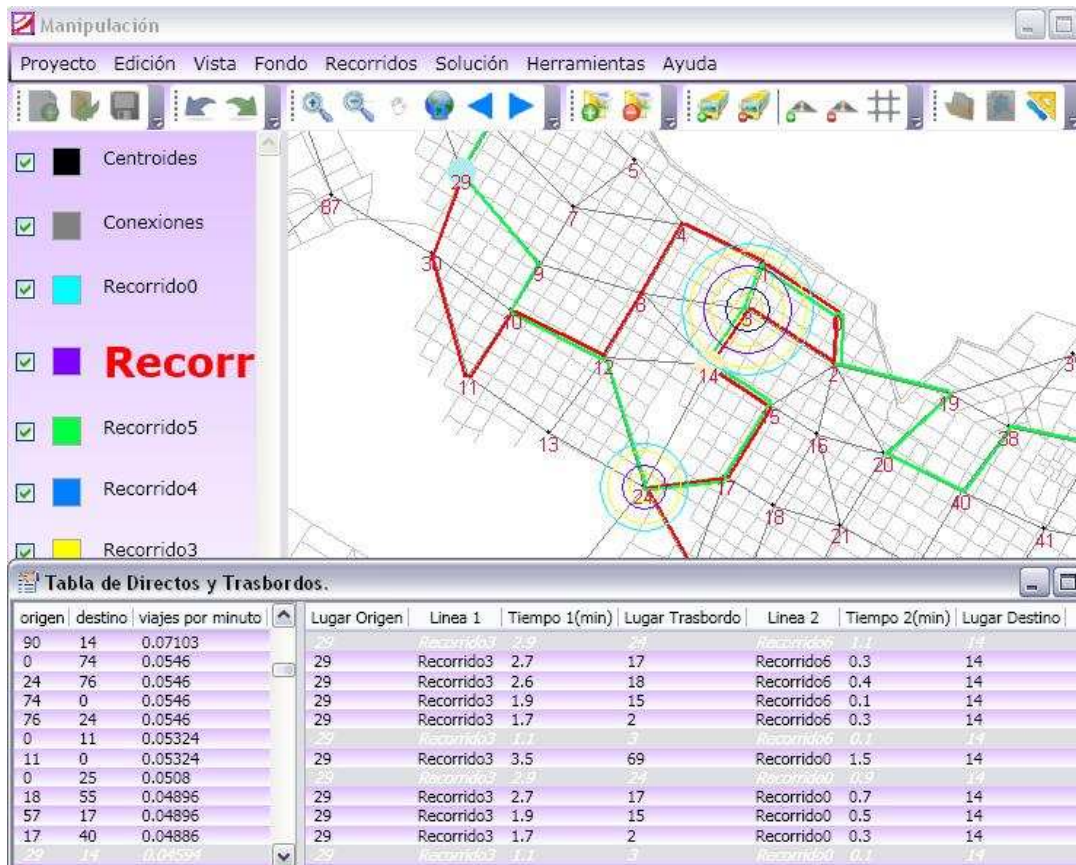


Figura 5.2.2.5. Esta captura muestra al usuario utilizando la herramienta “directos y trasbordos”. Esta herramienta despliega la demanda de los nodos origen destino ordenadas de mayor a menor (tabla en esquina inferior izquierda de la imagen). El usuario en este caso seleccionó el origen 29 y destino 14. Al seleccionar, se despliega en pantalla (al costado de la tabla mencionada antes) una tabla que contiene los trasbordos que se pueden realizar utilizando los recorridos de la solución para ir desde el origen al destino. En el mapa se visualizan los recorridos que sirven para ir desde el nodo origen al nodo destino de forma directa (sin trasbordos), en este caso el recorrido 5 (verde) y el recorrido 2 (rojo). Además, también se puede visualizar los trasbordos como círculos concéntricos en el lugar del traslado. En este caso el usuario ha seleccionado algunos trasbordos, en particular los que utilizan como lugar de traslado el nodo 24 y el 3.

6. Anexo. Búsqueda y Selección de Tecnologías SIG

6.1 Introducción

Este documento es producto de la investigación que se llevó a cabo para involucrarse con la temática que presentaba este proyecto. El mismo no intenta ser una guía en cuanto a los temas que se tratan, sino que el objetivo es salvar el conocimiento obtenido durante el tiempo de investigación, la cual se enfocó en los sistemas de información geográfica libres.

Sistemas de Información Geográfica

Un sistema de información geográfica (SIG) es una herramienta tecnológica para comprender la geografía y así tomar decisiones inteligentes. Integra software, hardware y datos para capturar, analizar y visualizar toda la información que tenga referencia geográfica. Los SIG permiten visualizar, entender, cuestionar e interpretar de varias formas los datos y de esta manera, revelar relaciones, patrones y tendencias en forma de mapas, reportes y gráficos. Los SIG organizan la información para que los usuarios leyendo mapas puedan seleccionar la información pertinente a la tarea. Un mapa temático tiene una tabla de contenidos para que los usuarios puedan sumar nuevas capas a un mapa base con información geográfica de lugares reales del mundo.

Un buen programa SIG permite procesar datos geográficos de distintas fuentes e integrarlas para formar un mapa de proyecto.

Los mapas SIG son interactivos. En la pantalla de la computadora los usuarios pueden explorar el mapa en cualquier sentido, acercarse o alejarse, y cambiar la naturaleza de la información contenida en el mapa.

Esta tecnología permite contestar preguntas y resolver problemas mediante la observación de la información la cual se muestra fácil de comprender y compartir.

Referencias [GIS 2008], [EGG 2008].

6.2 SIG-T

Un sistema de información geográfica para el transporte (SIG-T) se refiere a la aplicación de tecnología de información y conocimientos relacionados con problemas del transporte.

El transporte y la geografía son sistemas complejos altamente relacionados. El transporte existe para sobreponer las discrepancias geográficas de los recursos, bienes y servicios, moviendo materiales, personas o información que indica donde están y donde se desea que se encuentren. El transporte es también un componente relevante en la calidad de vida y el desarrollo sostenible.

Los analistas y encargados de la toma de decisión del transporte utilizan las herramientas SIG para tareas como planeamiento de infraestructura, diseño y manejo, planeamiento del tránsito y operaciones, análisis de tráfico y control, mitigación de riesgos, y configuración y manejo de sistemas complejos de logística, dentro de muchas otras.

Las herramientas SIG-T que existen en la actualidad son una combinación de funcionalidades de los sistemas de información geográfica y distintos algoritmos y modelos referentes a los problemas de transporte [OW 1990].

6.2.1 Funciones de análisis de los SIG-T

Existe un conjunto de funciones de análisis básicas de los SIG que se encuentran en casi cualquier software SIG. Estas funciones son útiles para muchas aplicaciones SIG, incluyendo las aplicaciones SIG-T. Esta sección esta basada en [MS 2001].

6.2.1.1 Básicas.

Consulta

Una función fundamental para soportar el análisis en los SIG-T es la habilidad de consultar las bases de datos SIG según los criterios especificados por los usuarios.

Superposición

La superposición ha sido utilizada como una herramienta de análisis de mapa mucho antes de la era del SIG. La superposición se maneja de forma diferente en los SIG raster que en los SIG vectoriales. Un SIG raster divide al mapa en una grilla regular. Cuando dos mapas se superponen para obtener nueva información, los valores correspondientes a las celdas de la grillas en los dos mapas son evaluadas. En los SIG vectoriales se asume un sistema de coordenadas continuo. Requiere comparar las coordenadas de los elementos de los dos mapas para evaluar las relaciones topológicas.

Buffer

La operación de buffer se utiliza cuando se necesita hallar los elementos que están localizados a una distancia específica de otro en el mapa. La operación de buffer puede ser llevada a cabo sobre puntos, líneas o polígonos. Aplicar la operación buffer sobre un punto genera un círculo de radio igual a la distancia especificada en la operación. En el caso de una línea, la operación crea un polígono que se extiende la distancia especificada por el usuario. Para un polígono, el usuario tiene la opción de crear una zona de buffer que se extienda hacia dentro o hacia fuera del polígono, o crear zonas buffer en las dos direcciones. En las tres situaciones mencionadas, la operación de buffer automáticamente disuelve los bordes en las áreas superpuestas entre zonas de buffer adyacentes. Los polígonos creados por la operación de buffer se generan en nuevas capas que pueden ser usadas por otras capas para análisis posteriores como superposición topológica.

Spatial Join

Esta operación está basada en las relaciones espaciales entre los elementos de dos capas diferentes. Para las relaciones espaciales entre punto-punto, punto-línea y línea-punto el spatial join encuentra el punto o línea más cercana en la segunda capa a cada elemento de la primera capa. La distancia utilizada para el spatial join es la distancia Euclídea esto puede ser inadecuado para las aplicaciones SIG-T que requieren distancia de red.

Spatial join basada en otro par de primitivas geométricas está fundamentada en las relaciones topológicas. Spatial join entre línea-línea determina si una línea en una capa es parte de alguna línea en la otra capa y une los registros empatados. Cuando se aplica spatial join entre las relaciones espaciales de punto-polígono, línea-polígono, y polígono-polígono, los elementos de la primera capa que se encuentran completamente contenidos en los elementos de la segunda capa son identificados.

6.2.1.2 Funciones específicas de análisis para transporte

Camino más corto y ruteo

El problema del camino más corto consiste justamente en encontrar el camino más corto entre dos nodos de una red tal que la suma de las aristas ponderadas del camino sea mínima. Existen variantes según la cantidad de unidades de flujos a través de la red. Las soluciones más implementadas en los SIG, consideran una única unidad

de flujo. Cuando se rutean varias unidades de flujo, se deben tener en cuenta las alternativas externas para rutear como la congestión, en otras palabras, el efecto de escoger un camino sobre los demás. El problema del camino más corto es muy útil para muchas aplicaciones. También están embebidos en otros más complejos como problemas de ruteo y flujo de redes.

Dos problemas relacionados entre sí que aparecen frecuentemente en los problemas de logística, o procuración y/o distribución de bienes y servicios son el problema del vendedor viajante (TSP – traveling salesman problem) y el problema de enrutamiento de vehículos (VRP – vehicle routing problem).

El TSP es uno de los problemas más famosos de optimización. El nombre viene de la analogía de un vendedor que debe visitar todas las ciudades de un determinado territorio y retornar a la ciudad de comienzo. Para modelarlo se asume una red donde cada lugar que debe visitar es un nodo (lugares de demanda) y los arcos representan conexiones directas entre los lugares de demanda. Si se está resolviendo el TSP en espacio plano, los arcos representan los viajes directos entre los pares de lugares de demanda y los pesos de los arcos corresponden a la distancia entre los pares de lugares de demanda.

El TSP pertenece a una clase especial de problemas conocidos como NP-completo. NP-completo quiere decir dos cosas. Primero que el problema está en una clase en la cual no se conoce ningún algoritmo que pueda encontrar una solución en tiempo polinomial, lo que significa que para encontrar una solución se requiere tiempo exponencial. Sin embargo, y de alguna manera intrínseca, una hipotética respuesta puede ser verificada en tiempo polinomial. Segundo el término completo significa que el problema en algún sentido es equivalente al resto de los problemas NP. Si se resuelve el TSP de forma óptima en tiempo polinomial se pueden resolver todos los problemas NP de forma óptima en tiempo polinomial.

El VRP es esencialmente la versión del TSP con flota. El problema es rutear un número fijo de vehículos de carga o de descarga a través de un número de lugares de demanda de manera que el costo sea mínimo y se cumplan las restricciones de capacidad de los vehículos. Típicamente, existe un lugar designado como depósito, donde todos los vehículos deben empezar y terminar sus recorridos. El depósito usualmente no es miembro del conjunto de lugares de demanda.

El VRP es un problema NP-difícil. Un problema es NP-difícil si no es NP pero cualquier problema NP puede ser convertido a él. No NP significa que el problema es más difícil que los problemas NP, por ejemplo no se puede verificar una hipotética respuesta en tiempo polinomial. Esto quiere decir que NP-difícil es difícil o más difícil que cualquier problema NP, lo que significa, que no se conoce ninguna solución de algoritmo en tiempo polinomial.

Modelos de interacción espacial

Los modelos de interacción espacial estiman la cantidad de interacción entre ubicaciones geográficas. Ejemplos de interacción son el movimiento de personas, de materiales, de energía o de información. Estos modelos consideran los factores que generan los viajes en los orígenes, los atractivos de los destinos, y los costos de los viajes entre los pares orígenes-destinos. Los modelos de interacción son fundamentales para la previsión de la distribución de los viajes y también para comprender mejor los modelos de transporte de demanda de viajes y utilización de tierra. Una característica particular de los modelos de interacción es que la información acerca de la interacción corresponde a pares. Un modelo de datos efectivo para representar este tipo de información es la matriz de flujo. Esto es un problema para los manejadores de bases de datos implementados en los SIG comerciales dado que no están diseñados para manejar información asociada con pares. Algunos paquetes de SIG comerciales incluyen una extensión para acomodar estos pares de información en las tablas relacionales.

Modelos de flujos de redes

Existen dos amplias clases de problemas para considerar en los modelos de flujos de redes. El primero es cuando la red no se puede congestionar. Esto significa que el costo de una unidad de flujo no afecta otros flujos, en otras palabras, no hay que tener en cuenta la interacción y la congestión entre flujos. Este caso usualmente corresponde a redes abstractas que representan problemas en logística y en la procuración y distribución de materiales y bienes a través del espacio. Estas redes abstractas representan posibles envíos, horarios, u otras relaciones lógicas que no necesariamente corresponden con la red de transporte físico. Ejemplos son problema de transporte, problema de flujo de costo mínimo y problema de flujo máximo.

El segundo caso involucra a redes que se pueden congestionar. En este caso, las redes se saturan al punto que los flujos comienzan a afectarse unos a otros, creando

demoras y otros costos relacionados con la congestión. Esto usualmente corresponde con flujos a través de redes físicas como calles y otras redes de transporte. Ejemplos son el equilibrio del usuario óptimo y el equilibrio del usuario óptimo y dinámico

Los modelos de redes de flujo han sido un debe en las aplicaciones SIG, con muy pocas excepciones (TransCAD).

Modelos de ubicación de instalaciones

En su forma más general, los modelos de ubicación de instalaciones determinan la cantidad de instalaciones, su localización y la asignación de demandas entre éstas. Casos especiales de este modelo general, fijan una o más variables y buscan distintos objetivos. Por ejemplo se fija el número de instalaciones y se minimiza la suma de los costos de viajes, cubriendo todos los puntos de demanda dentro de una cierta distancia.

Existen tres tipos de problemas de ubicación de instalaciones basados en sus objetivos. El problema de la mediana (median problem) intenta localizar las instalaciones de manera de minimizar el costo de viaje total (o, de manera equivalente, el promedio) entre los lugares de demanda y las instalaciones. Este es el objetivo aplicado más comúnmente. Es apropiado para instalaciones del sector privado como puntos de venta e instalaciones del sector público como servicios del gobierno. El problema del centro (center problem), también conocido como problema minimax, localizan las instalaciones de manera de minimizar el costo de viaje máximo entre los puntos de demanda y las instalaciones. Este es aplicado generalmente en los servicios de emergencia donde la preocupación es minimizar el peor tiempo de respuesta o costo en el sistema. El problema de requisito (requirement problem) localiza la instalación según algún estándar de rendimiento especificado. Por ejemplo se puede definir el servicio a un lugar de demanda según una distancia especificada a una instalación.

6.2.1.3 Ejemplos de SIG-T

TransCAD. Extraído de [CA 2009]

TransCAD es un SIG producido por Caliper Corporation, diseñado para técnicos del transporte de manera que puedan almacenar, visualizar, manejar y analizar información referente al transporte. Combina capacidades de los SIG y de los modelos de transporte en una única plataforma integrada. Puede ser utilizado para distintos modos de transporte a distintas escalas y niveles de detalle. TransCAD provee: un

potente motor SIG con extensiones especiales para el transporte; herramientas de mapeo, visualización y análisis diseñadas específicamente para aplicaciones de transporte; módulos para previsión de demanda de viajes, transporte público, logística, ruteo, localización, y manejo territorial. Los módulos de aplicación en TransCAD están completamente integrados con funcionalidades SIG para de esta manera mejorar el rendimiento y hacer más sencillo el uso.

TransCAD extiende el modelo de datos tradicional de los SIG para incluir objetos como: redes de transporte, matrices, rutas, datos linealmente referenciados.

Las redes de transporte son estructuras de datos especializadas utilizadas para modelar flujos dentro de la red. Incluye características especiales como: penalizaciones o restricciones por desvíos; puentes, túneles y enlaces de un sentido; intersecciones y uniones; terminales, trasbordos, y funciones de retraso; enlaces entre centroides de zonas; clasificaciones de enlaces y funciones de rendimientos; accesos, salidas, y enlaces de transferencia a pie.

Las matrices son esenciales para muchas aplicaciones de transporte, pueden ser de distancia, de tiempos de viaje, flujos origen destino. TransCAD provee funcionalidades para crear y manipular matrices, y herramientas para análisis espacial y visualización avanzada.

Las rutas y sistemas de rutas indican los caminos tomados por camiones, trenes, buses, o viajeros individuales de un lugar a otro. TransCAD provee funcionalidades para crear, visualizar, editar y manipular rutas. Se puede organizar a un conjunto de rutas en un único sistema de rutas y asociarle atributos, paradas, y programas de vehículos.

Los datos linealmente referenciados identifican la ubicación de un objeto de transporte como una distancia desde un punto fijo a lo largo de una ruta.

TransCAD es útil para muchos tipos de aplicaciones de transporte como: análisis de redes – implementa rutinas de camino mas corto, partición de redes, modelos TSP; planeamiento de transporte y modelado de la demanda de viajes – implementa modelos de generación/atracción/balanceo/distribución de viajes, elección modal, asignación; análisis de transporte; enrutamiento de vehículos y logística – enrutamiento y despacho de vehículos, enrutamiento sobre arcos, flujo de redes y análisis de distribución; manejo territorial y localización.

Cube. Extraído de: [CI 2009]

Cube, caballo de batalla de la empresa Citilabs (TP+, TRIPS, TRANPLAN), es una familia de productos de software que forman un completo sistema de pronóstico de demanda de viajes que proveen capacidades para facilitar el planeamiento de los sistemas de transporte. Cube está compuesto por Cube Base y bibliotecas complementarias que proveen funciones para el planeamiento. La ventana SIG de Cube Base utiliza una versión incorporada de ArcGIS de ESRI. Cube almacena todos los datos de la red en una geodatabase ESRI. Los usuarios de Cube tienen Cube Base y una o más bibliotecas complementarias dependiendo de la tarea de planeamiento que deben desarrollar. Cube Voyager provee funciones para pronosticar la demanda de viajes, Cube Cargo para el modelado de transporte de mercancías, Cube Dynasim para micro simulaciones multi modos, Cube Analyst para estimar y optimizar matrices de viajes a partir de conteos de tráfico y otros datos de campo, Cube Land para pronosticar el uso del suelo y el mercado inmobiliario.

Cube Voyager proporciona una completa biblioteca de funciones para el modelado y análisis de sistemas de transporte de pasajeros: calles, transporte público, peatones y bicicletas. Cuenta con diferentes módulos como redes, carreteras, transporte público, y demanda de viajes.

El módulo de redes permite crear representaciones detalladas de calles, intersecciones y rampas. Permite estimar rutas entre dos puntos, así como tiempos de viaje, distancias y costos. Permite representar intersecciones controladas señalizadas y no señalizadas, círculos y rampas como parte del proceso de distribución y de asignación.

El módulo de carreteras es utilizado para estimar los caminos entre zonas, y matrices de impedancia para ser empleadas en la accesibilidad. Los caminos se pueden construir con distintas metodologías como: todo o nada, todos los caminos más corto, Dial, Burrell. También se pueden agregar restricciones en las intersecciones y/o en los tramos. La asignación es iterativa e incremental, utilizando ajuste de equilibrio.

El módulo de transporte público provee funcionalidades avanzadas para los sistemas de transporte público. Posee procesos automáticos para crear tramos para los distintos modos y para trasbordos entre servicios. Los caminos entre dos puntos se encuentran utilizando distintas técnicas como todo o nada, multitrayectoria estocástica o

multitrayectoria en ruta discreta. Se pueden representar niveles de servicios utilizando intervalos u horarios de transporte público. Permite representar estructuras de tarifas complejas y modelar restricciones de capacidad para el transporte público.

El módulo de demanda de viajes procesa datos zonales y matrices de acuerdo a expresiones especificadas por el usuario. La entrada son los datos zonales y las matrices y la salida es representada por matrices y reportes. Permiten utilizar distintos tipos de procesos de demanda como modelos de elección jerárquicos y multinomiales, análisis por categoría y modelos de regresión, modelos de gravedad, métodos de fratar.

Cube Voyager utiliza una estructura modular y basada en un lenguaje de script que permite incorporar cualquier metodología en los modelos, desde los convencionales de cuatro etapas hasta los de elección discreta y enfoque basados en actividades.

Los modelos elaborados con Cube Voyager pueden ser de muchas formas: modelos de cuatro etapas - incluye plantillas para elaborar estructuras de generación, distribución, elección modal y asignación; modelos modificados de cuatro etapas con retroalimentación – incluye modelos de posesión de vehículo, elección modal y de destino combinado; demanda basada en actividades; modelos de equilibrio combinados.

6.3 SIG Libre

Licencias de Software

En esta sección se trata el tema del software libre. Está en relación a uno de los principales requisitos de este proyecto, el cual especifica que el software geográfico a utilizar para la construcción de la herramienta debía ser no propietario. El grupo conocía solamente el término software libre pero no sabía en concreto que implicaciones podría tener esto sobre el trabajo. De ahí surgió la necesidad de conocer un poco más acerca de las licencias de software. Esta sección contiene una breve descripción de los principales conceptos, para entender un poco más el tema.

Primero se definirá el término licencia. Según la Real Academia Española una licencia es “permiso para hacer algo, abusiva libertad de decir u obrar,...”.

Llevando el concepto de licencia al tema del software, una licencia de software es la autorización o permiso otorgado por el autor para utilizar su trabajo de una forma convenida según los límites y derechos establecidos con respecto a su uso. Posibles restricciones establecidas (en lo que respecta al software) pueden ser plazo de cesión de

los derechos, restricción del territorio, límites en la responsabilidad por fallos, o cualquier otra condición que el autor del software desee incluir.

Existen distintas categorizaciones de las licencias de software. Quizás la más general es dividir las licencias de software según el tipo de software que cubren. Las más notables son las licencias de software propietario y las licencias de software libre.

Se utiliza el término software propietario para referirse a todo aquel software que limita a los usuarios en cuanto a las posibilidades de usarlo, modificarlo o redistribuirlo, o cuyo código fuente no está disponible o el acceso se encuentra restringido.

A su vez, se utiliza el término software libre para definir cualquier software que le brinde al usuario la libertad de ejecutar, copiar, distribuir, estudiar y modificar. Tanto para estudiar como para modificar es necesario el acceso al código fuente.

6.4 *Software libre*

6.4.1 Movimientos del software libre

6.4.1.1 FSF

El movimiento para el software libre fue creado por Richard Stallman al comienzo de los años 80 (1983), cuando lanzó el proyecto GNU. A mediados de los ochenta (1985) Stallman creó la Fundación para el Software Libre (Free Software Foundation). Una fundación sin fines de lucro con la misión de promover y educar en el uso del software libre en todas las áreas de la computación. Esta fundación se encarga en la actualidad de los asuntos legales, organizativos y de promoción que involucran a la comunidad de usuarios de software libre.

6.4.1.2 GNU

El objetivo del proyecto GNU era crear un sistema operativo completo tipo Unix de software libre. El núcleo de GNU no se terminó así que se utiliza el núcleo Linux. Esta combinación es el conocido sistema operativo GNU/Linux más conocido simplemente como Linux. La principal organización que patrocina el proyecto es la FSF. El proyecto GNU apoya los objetivos de la FSF de preservar, promover y proteger la libertad de uso, estudio, copia, modificación y redistribución del software.

6.4.1.3 Open Source

La iniciativa para el código abierto (The Open Source Initiative - OSI) es una corporación sin fines de lucro de California fundada en 1998. Fue creada para enseñar y proteger los beneficios que brinda el código abierto y para crear puentes entre las distintas divisiones dentro de la comunidad del código abierto. Una de las actividades más importantes que lleva a cabo la OSI es la de establecer estándares donde se mantenga viva la definición del código abierto (The Open Source Definition - OSD). La OSD representa la filosofía del código abierto y también define los límites del uso, modificación y redistribución del código abierto.

6.4.2 Licencias del software libre

Las licencias de software libre se suelen dividir en dos tipos licencias copyleft y licencias permisivas.

6.4.2.1 Licencias Copyleft

Las Licencias Copyleft comprenden un grupo de derechos de autor que se caracterizan por eliminar las restricciones de distribución o modificación. Nace como contraposición al copyright. Se lo divide generalmente en dos subgrupos copyleft fuerte y copyleft débil. El copyleft fuerte hace referencia a la eficiencia con la que hace cumplir las condiciones de la licencia a todos los tipos de trabajos derivados. El copyleft débil hace referencia a las licencias que no se heredan a todos los trabajos derivados.

Dentro de las licencias copyleft encontramos:

GNU GPL

La licencia GPL es una licencia de software libre que fue desarrollada por la Free Software Foundation (FSF). El objetivo de esta licencia es proteger la libre distribución, modificación y uso de software.

Protege a los programas que se encuentran bajo la licencia de los intentos de apropiación que limiten a los usuarios. La licencia GPL asegura su continuidad ya que cualquier software que utilice un software bajo esta licencia queda cubierta por ella. Esta licencia es una licencia copyleft fuerte.

GNU LGPL

La licencia LGPL es una licencia de software libre que al igual que la GPL fue desarrollada por la FSF. Se diseñó como un compromiso entre las licencias copyleft y las licencias permisivas. La licencia LGPL coloca las restricciones copyleft en el programa pero no en cualquier otro software que utilice el programa. Esta licencia es muy utilizada en las librerías de software. La LGPL es una licencia copyleft débil.

MPL

La licencia MPL es una licencia de software libre que fue desarrollada por Netscape Communications Corporation una división de la compañía American Online, y más tarde fue traspasado a la Fundación Mozilla. La licencia MPL cumple con los postulados de la OSI y la FSF. Esta licencia sin embargo permite la reutilización no libre del software sin restringir la reutilización del código ni relicenciar bajo la misma licencia. Esta licencia es una licencia copyleft débil.

6.4.2.2 Licencias Permisivas

Las Licencias Permisivas comprenden el grupo de licencias que no imponen condiciones especiales, sólo especifican que el software se puede redistribuir y modificar.

Dentro de las licencias permisivas encontramos:

BSD

La licencia BSD es una licencia de software libre que fue desarrollada por la Universidad de California en Berkeley para el sistema operativo Berkeley Software Distribution (BSD), de ahí deriva su nombre. Esta licencia mantiene el derecho de autor para la posible renuncia de garantía y para el posible requisito de atribución de autoría en los trabajos derivados. Permite la libre redistribución (incluso como no libre) y modificación.

MIT

La licencia MIT es una licencia de software libre que fue desarrollada por el MIT (Massachusetts Institute of Technology). Esta licencia puede ser modificada para necesidades particulares. Permite la reutilización del software tanto para el software libre como para el software propietario, de esta manera uno no tiene necesidad de liberar los cambios realizados al programa original. Además permite licenciar los cambios utilizando otra licencia. Es similar a la licencia BSD.

Referencias [FSF 2008], [GNU 2008], [OSI 2007], [MG 2007], [EA 2008], [SL 2005], [GL 2005], [OGC 1994], [OSG 2008].

6.5 *Software Analizado*

6.5.1 Herramientas

Algunos sitios Web relevantes en lo que refiere a los SIG y sobre todo a SIG libre son los siguientes.

OGC – Extraído de [OGC 1994]

La OGC (Open Geospatial Consortium) es una organización internacional sin fines de lucro cuyo fin es el desarrollo de estándares para los servicios GEO espaciales y los servicios basados en localización. Este consorcio industrial internacional nuclea a compañías, agencias de gobiernos y universidades para lograr un consenso en lo que refiere a la especificación de interfaces para los servicios mencionados.

La especificación OpenGIS es una marca registrada de la OGC y es su nombre de marca asociado con todos los documentos y las especificaciones que esta produce. Esta especificación faculta a los desarrolladores de tecnología para hacer información y servicios espaciales complejos accesibles y útiles para todo tipo de aplicaciones.

OSGeo – Extraído de [OSG 2008].

Este sitio es mantenido por la fundación para el código abierto GEO espacial. Esta fundación intenta apoyar la creación de programas GEO espaciales de alta calidad. El sitio sirve como portal para que tanto usuarios como desarrolladores compartan y contribuyan en el desarrollo de proyectos.

El sitio contiene varios proyectos de distintos tipos los cuales se encuentran en diferentes secciones: WEB, aplicaciones de escritorio, librerías GEO espaciales, catálogos de metadata y otros. Para la herramienta a construir fueron de interés las aplicaciones de escritorio y librerías GEO espaciales. En aplicaciones de escritorio se encontraron los siguientes proyectos GRASS GIS, OSSIM, Quantum GIS y gvSIG. Y en bibliotecas GEO espaciales FDO, GDAL/ORG, GEOS y GeoTools.

FreeGIS – Extraído de [B 2008]

Este sitio es mantenido por el Proyecto FreeGIS. Este intenta promover el uso desarrollo y el soporte de los SIG libres, así como también toda la información geográfica.

El sitio contiene una cantidad importante de programas, proyectos, información y documentos, todos relacionados por supuesto con los SIG.

OpenSourceGIS – Extraído de [OS 2008]

Esta página intenta ser un índice que facilite el acceso y utilización de las distintas herramientas SIG de código abierto y libre. Se trata de centralizar la gran cantidad de información existente, sobre el software orientado a este tipo de aplicaciones, tanto para escritorio como para Web.

Basados en las opciones existentes en la Web, sobre todo en las mencionadas, se comenzó a conocer un poco más las funcionalidades brindadas por algunas de las herramientas. A continuación se encuentran algunas de las que fueron probadas separadas en dos categorías, aplicaciones de escritorio y bibliotecas (probadas significa conocimiento de las funcionalidades que brinda, instalación y ejecución en el caso de aplicaciones de escritorio, conocimiento de las funcionalidades en el caso de librerías).

6.5.2 Aplicaciones de escritorio.

6.5.2.1 MapWindow

MapWindow GIS- Extraído de [MW 2008]

El proyecto comienza en 1998 por miembros del Laboratorio de Investigación del Agua de Utah (Water Research Lab) en la Universidad Estatal de Utah, y la idea inicial fue construir un componente básico GIS que proveyera las funcionalidades para desarrollar aplicaciones GIS personalizadas.

MapWindow consiste en la aplicación principal de MapWindow, componentes del núcleo, y plug-ins.

La aplicación principal de MapWindow es la interfaz central. Permite visualizar mapas vectoriales y mapas raster, así como imágenes, todo esto en distintos formatos.

Los componentes del núcleo son los que operan por debajo de MapWindow. Estos son tres, MapWinGIS, MapWinInterfaces y MapWinGeoProc.

MapWinGIS es un control ActiveX que puede ser utilizado por cualquier lenguaje de programación que soporte ActiveX. Este es el componente principal del núcleo. Esta construido para ser un modelo de interfaz con todas las funcionalidades no sólo como un visualizador de mapas. Esto implica acelerar la subida de imágenes y el despliegue de grillas, limitar la cantidad de dibujado que el usuario observa, e incluir una API para el acceso de bajo nivel a grillas, elementos, tablas y datos de imágenes.

MapWinInterfaces es un archivo DLL que permite a los usuarios escribir sus propios Plug-ins para la aplicación MapWindow.

MapWinGeoProc es una librería para el procesamiento geográfico.

Los Plug-ins son herramientas especializadas para interactuar con la aplicación principal de MapWindow. La aplicación principal de MapWindow es principalmente una herramienta para visualizar datos, la verdadera potencia de MapWindow se encuentra en los plug-ins. Dentro de varios plug-ins implementados, cuatro vienen con la aplicación principal de MapWindow, Editor de Tablas, Editor de Shapefiles, Herramientas GIS e Identificador/Etiquetador de Elementos.

Las funcionalidades que ofrece el núcleo de MapWindow varían según el tipo de datos sobre las que son aplicadas, raster, vectores e imágenes.

Las funcionalidades raster operan principalmente sobre grillas de varios formatos como USU Binary Grids, ESRI Grids, y GeoTIFF grids. Permiten asignar proyecciones a objetos raster, las cuales son escritas en la grilla pero sin que ocurra una reproyección actual. Realiza reproyecciones para pasar los datos de una grilla a datos en otro tipo de proyección. Puede cambiar el formato de las grillas. Crea imágenes GEO referenciadas a partir de una grilla. Permite fusionar varias grillas en una. Recorta grillas con polígonos. GEO referenciar imágenes mediante la especificación de tres puntos sobre esta.

Las funcionalidades vectoriales son llevadas a cabo sobre datos vectoriales como ESRI shapefiles. Permite al igual que sobre raster asignar proyecciones a capas y permite también reproyectarlas. Lleva a cabo buffers de elementos. Calcula el área de los polígonos. Recorta polígonos de una capa con líneas de otra. Permite recortar una capa con polígonos y generar huecos en capas también mediante el uso de polígonos.

Exporta los elementos seleccionados a una nueva capa. Permite aquí también la fusión de varias capas sobre una y la fusión de varios elementos en uno.

Por último sobre las imágenes, estas funcionalidades trabajan con datos en algún formato de imagen. Permite asignar proyecciones a las imágenes, rectificarlas a WorldFile y reproyectarlas.

Documentación:

La documentación para la aplicación MapWindows cuenta con unos cuantos buenos tutoriales en formato texto y documentación online. Además cuenta con algunos videos mostrando algunas de las funcionalidades que éste ofrece.

La documentación para MapWinGis cuenta con algunos tutoriales y con documentación online. Los tutoriales de MapWinGis son cortos, sólo se muestran algunas de las funcionalidades. Además cuenta con unos cuantos ejemplos.

Tiene especificadas todas las funcionalidades en un texto que se actualiza periódicamente.

6.5.2.2 GRASS

GRASS GIS- Extraído de [GG 2009]

GRASS (Geografic Resources Analysis Support System), originalmente fue desarrollado por el cuerpo de ingenieros de la armada de Estado Unidos, como una herramienta para el manejo territorial y el planeamiento del ambiente.

Es un SIG utilizado para el manejo de información, procesamiento de imágenes, producción de gráficos, modelado espacial, y visualización de muchos tipos de datos. Es un software libre de código abierto bajo la licencia GNU/GPL. Es un proyecto oficial de la OSGF.

GRASS contiene más de 350 programas y herramientas para "renderizar" (desplegar) mapas e imágenes en monitor o en papel, manipula información raster, y vectorial, procesa imágenes multiespectrales, crea, maneja y guarda información espacial.

GRASS soporta el trabajo grupal a través del concepto LOCATION/MAPSET, el cual puede ser configurado para ser compartido sobre NFS. Un location contiene datos de una extensión geográfica en un mismo sistema de coordenadas. Cada location tiene un

directorio PERMANENT que almacena la información básica acerca del location. Un MAPSET es una organización de mapas según algún criterio, como tema, proyección, etc. Es un subconjunto de un location, técnicamente son subdirectorios bajo estos. Cada sesión de GRASS corre bajo el nombre de un mapset. Juegan un papel especial cuando muchos usuarios trabajan a la vez sobre un mismo location. La información de los mapsets para un location dado puede ser leída por cualquiera pero modificada solo por quien posea permiso.

GRASS esta escrito en ANSI-C y es conforme al estándar POSIX.

Sistemas Operativos:

GNU/Linux, Solaris, SGI IRIX, HP UX, MAC OSX, IBM AIX, UNIX BSD, Free BSD, CRAY Unicos, MS-Windows nativo o Gygynus.

Procesamiento de datos vectoriales:

GRASS permite importar datos vectoriales de dos maneras. Una utiliza la librería OGR la cual permite la cargar datos de diferentes formatos. La otra manera permite la carga desde archivos de texto o desde una base de datos, ambos contienen coordenadas y datos de atributos. Con respecto a la exportación GRASS permite exportar los datos en varios formatos como, DXF, formatos soportados por OGR (escritura), mapa de vector GRASS ASCII, POV-Ray, SVG, VTK ASCII.

Cada mapa guarda información general como atributos y un historial. El historial almacena los comandos que han sido utilizados para la formación del mapa. GRASS permite visualizar esta información y además actualizar datos como titulo, escala, organización de esta metadata del mapa.

Las operaciones vectoriales sobre un mapa siempre son llevadas a cabo sobre todo el mapa. Si lo que se desea es operar sobre una parte del mapa y no sobre todo, se deben aplicar operaciones de recorte para obtener la parte deseada.

GRASS es un GIS topológico. Esto quiere decir que los componentes geográficos de un mapa están relacionados. A manera de clarificar, en un SIG no topológico si dos áreas poseen un borde común este borde se digitaliza dos veces una para cada área y además este borde es almacenado dos veces. En cambio en un SIG topológico el borde existe y es compartido. A la información topológica en GRASS se la denomina nivel 1, a la información no topológica, también conocida como espagueti, se la denomina nivel 2.

GRASS maneja (hasta el momento) cinco tipos de objetos vectoriales:

point: un punto.

line: una secuencia directa de vértices conectados con dos puntos extremos llamados nodos.

boundary: la línea borde que describe un área.

centroid: un punto dentro de un boundary.

area: la composición topológica de centroid y boundary.

Muchas operaciones como consultas, superposición, exportación, se realizan con objetos que tienen asignado un número de categoría. El número de categoría es el identificador del vector. Con este identificador se asocian los atributos del vector. Un objeto vectorial puede tener cero o más categorías. Estos números se almacenan tanto en el archivo de la geometría como en la tabla de los atributos para cada objeto vectorial. Para asociar un objeto vectorial a varias tablas de atributos este debe poseer tantos números de categorías como tablas se quieran asociar.

Es posible asociar los objetos geográficos de un mapa con una o más tablas. Cada asociación con una tabla de atributos distinta es llamada capa. Las asociaciones definen el driver de la base de datos, la base de datos y la tabla, a ser utilizadas. Los números de categoría de un archivo geométrico corresponden a las filas en la tabla de atributos. Las capas contienen las asociaciones a las tablas de atributos, no contienen objetos vectoriales. De esta manera algunos objetos geométricos pueden aparecer en algunas capas y en otras no. Esto permite realizar agrupaciones de objetos vectoriales según temas sin perder las relaciones topológicas.

Se pueden utilizar dos drivers de base de datos DBF y DBMS. DBF provee un soporte SQL muy limitado mientras que DBMS provee un soporte SQL total ya que las consultas son enviadas directamente a la interfaz del manejador.

Operaciones sobre mapas: segmentación de líneas, buffer de las distintas geometrías, reproyección de mapas, mapas sin proyección pueden ser GEO codificados, conversiones de triangulación y punto a polígono, generación de puntos aleatorios, superposiciones de mapas, selecciones, obtener estadísticas, convertir información vectorial a base de datos y a raster, consultas, pasar información raster a mapas vectoriales.

GRASS provee soporte para el análisis de redes vectoriales. Implementa los siguientes algoritmos: mantenimiento de redes, camino más corto, TSP, creación de subredes, árboles mínimos de Steiner, iso-distancias.

Permite crear sistemas de referencia lineal y consultarlo.

Procesamiento de datos raster:

Los límites geográficos de un mapa raster están descritos por los campos norte, sur, este, y oeste. Estos valores describen las líneas que delimitan el mapa. Como regla general los mapas raster deben cumplir lo siguiente, los mapas raster de salida deben tener los límites y la resolución igual que aquellos de la región actual, además, los mapas raster de entrada son automáticamente ajustados y reescalados y son automáticamente enmascarados si existe un mapa raster llamado MASK.

GRASS utiliza la librería GDAL para la importación y exportación de mapas. Esta ofrece una interfase para diferentes formatos raster. También ofrece opciones como creación de location al instante o extensión de la región por defecto para ajustar la extensión del mapa raster importado. Siempre se importa el mapa completo.

GRASS permite ver la información general acerca de un mapa como por ejemplo, región de extensión, rango y tipo de datos, historia, y otra metadata. La metadata como título, unidades, datum vertical, etc., puede ser actualizada. Permite además manejar la región de extensión y la resolución.

En cuanto a las operaciones, estas siempre son aplicadas con las propiedades de la región actual, por ejemplo la extensión actual, la resolución actual. Si existe un mapa raster llamado MASK, la mayoría de los módulos raster de GRASS operan con los datos que caen dentro del área de la máscara, y tratan los datos que caen afuera como nulos. La máscara solo es aplicada cuando se lee un mapa raster. Esta máscara es cargada como un mapa de enteros. Si MASK es un mapa de reales, los valores son convertidos.

GRASS posee una serie de comandos para calcular estadísticas locales y globales. Se pueden generar perfiles, así como histogramas y diagramas polares. También genera reportes.

GRASS provee varios métodos algebraicos de muestreo e interpolación para mapas raster.

Proyecciones y transformaciones espaciales:

Un location en GRASS es referenciado con una única proyección y sistema de coordenadas. Cuando se crea un nuevo location a partir de un mapa raster o vectorial se deben definir la proyección y el sistema de coordenadas. Para cambiar la proyección de los mapas, un nuevo location debe ser creado y los mapas que se desea tener deben ser reproyectados en el nuevo location.

GDAL es una librería para traducir la información GEO espacial de rasters. Se encuentra bajo licencia X/MIT. Por ser una librería, presenta un solo modelo de información abstracta a la aplicación llamante para todos los formatos soportados. También viene con una serie de comandos de línea para traducir y procesar información.

OGR es una librería escrita en C++ de código abierto que provee acceso de lectura (y en algunos casos escritura) a una variedad de archivos de vectores de distintos formatos incluyendo ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, y Mapinfo mid/mif y formato TAB. Al igual que GDAL posee una serie de comandos de línea para la traducción y procesamiento de información. Es parte de la librería GDAL.

6.5.2.3 gvSIG

gvSIG: Extraído de [GV 2008]

Iniciado en el año 2003, es un proyecto de desarrollo informático cofinanciado por la Consejería de Infraestructuras y Transporte de la Generalidad Valenciana y la Unión Europea.

A finales del 2003 la CIT (Consejería Internacional del Transporte, Valencia-España) sacó a concurso el desarrollo e implantación de un nuevo programa para el manejo de información geográfica (SIG).

gvSIG (Generalitat Valenciana Sistema de Información Geográfico) es un programa para el manejo de información geográfica con precisión cartográfica que se distribuye bajo licencia GNU GPL. Permite acceder a información vectorial y raster así como a servidores de mapas que cumplan las especificaciones de la OGC. Esta es una de las principales características de gvSIG respecto a otros Sistema de Información

Geográfica, la importante implementación de servicios OGC: WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service), Servicio de Catálogo y Servicio de Nomenclátor.

El proyecto apuesta a ciertas características como, portabilidad, modularidad, código abierto, interoperabilidad, sujeto a estándares.

La arquitectura del proyecto tiene tres partes claramente diferenciadas: Una aplicación base que se puede extender mediante "plugins" y que se encarga de crear ventanas, cargar y gestionar extensiones, seleccionar el "look & feel", inicializar el idioma, etc. Es totalmente genérica, y sirve para crear nuevas aplicaciones MDI (multiple document interface).

Una librería de clases para la creación de aplicaciones GIS a medida. Dentro de esta librería encontramos clases para leer y escribir los formatos soportados, dibujar los mapas a las escalas adecuadas, asignar leyendas, definir simbologías, realizar búsquedas, consultas, análisis, etc.

Por último las extensiones a la aplicación base que contienen todo lo necesario para interacción con el usuario. Esta librería contiene los formularios para asignar leyendas, crear mapas, definir escalas, etc.

El propósito principal del proyecto era construir un cliente avanzado GIS de consulta, edición y creación de planos. Esta desarrollado completamente en JAVA. Es multilingüe. Algunas limitaciones de interés son la falta de documentación para desarrolladores y la masiva dependencia de librerías C++ y JAVA (ejemplo GDAL-OGR, GeoTools). Esto es un inconveniente dado que las dependencias son difíciles de evaluar y la programación personalizada se dificulta.

Cuando este SIG fue estudiado al comienzo del proyecto de grado, la herramienta no brindaba tantas facilidades para el desarrollador como hoy en día. No existía un manual para desarrolladores como actualmente tiene.

Se asistió a una charla dictada por la Ingeniera Victoria Agazzi representante de gvSIG , el día viernes 7 de noviembre en la Facultad de Ingeniería (UdelaR), en la cual mostraron que el emprendimiento había tomado mayor énfasis en los desarrolladores en los últimos meses. Al visitar el sitio nuevamente notamos estas diferencias.

En el momento que analizamos la herramienta, fue sencilla la parte de instalación de los binarios, es decir la aplicación para un usuario no desarrollador.

Pero la complicación mayor apareció cuando intentamos preparar el entorno de desarrollo. No fue posible ejecutarlo desde el código fuente. A continuación se mostrará un hilo de correos que se mantuvo con el foro de gvSIG, explicando esta situación. Vale destacar que el foro es muy activo, y las preguntas con sus respectivas respuestas se manejan en forma muy ágil, lo cual es otra ventaja de la herramienta.

De Jose Manuel Vivó (Chema) josemanuel.vivo@iver.es

Responder a Lista de Usuarios de gvSIG <gvsig_usuarios@runas.cap.gva.es>

Para Lista de Usuarios de gvSIG <gvsig_usuarios@runas.cap.gva.es>

fecha 8 de julio de 2008 3:54

Asunto Re: [Gvsig_usuarios] Problema al Compilar los Fuentes de GVSIG lista de distribución gvsig_usuarios.runas.cap.gva.es

Hola Diego:

Se queja de que no encuentra las clases que añade la librería Jai ("package javax.media.jai does not exist"). Se que has instalado esas librerías, pero el compilador no las encuentra. Lo único que se me ocurre es que al instalar estas librerías haya sucedido algún error. Desinstalas y vuelve a instarlas.

Espero que te sirva de ayuda.

Un Saludo.

Chema.

Respuesta por nuestra parte.

Hola José Manuel,

Lamentablemente sigo con el mismo problema.

Desinstalé las librerías JAI e instalé nuevamente pero ahora teniendo el cuidado que sean específicamente para jdk, es decir jai-1_1_3-lib-windows-i586-jdk.exe y jai_imageio-1_1-lib-windows-i586-jdk.exe.

Sólo son necesarias estas, no? es decir las jre no se necesitan, verdad?

Cuando corro el "Build All", las últimas líneas en consola muestran lo siguiente:

...

```
echo] Encoding: ISO_8859_1
```

```
javac] Compiling 276 source files to C:\gvsig-1_1_1-src\gvSIG-1_1_1-src\libCq CMS for java\bin
```

```
javac] C:\gvsig-1_1_1-src\gvSIG-1_1_1-src\libCq CMS for java\src\org\cresques\cts\gt2\CSGaussPt.java
```

```
:51: package javax.media.jai does not exist
```

```
javac] javax.media.jai.ParameterList params =
```

```
csFactory.createProjectionParameterList("Oblique_Stereographic");
```

```
javac] ^
```

```
javac] Note: Some input files use or override a deprecated API.
```

```
javac] Note: Recompile with -Xlint:deprecation for details.
```

```
javac] 1 error
```

```
BUILD FAILED
```

```
C:\gvsig-1_1_1-src\gvSIG-1_1_1-src\appgvSIG\build.xml:295: The following error occurred while executing this line:
```

```
C:\gvsig-1_1_1-src\gvSIG-1_1_1-src\libCq CMS for java\build.xml:60: Compile failed; see the compiler error output for details.
```

```
Total time: 1 minute 16 seconds
```

Es el mismo error que antes.

Como dice el leeme.txt seguí estos consejos:

"Para comprobar si lo tiene instalado compruebe que existe el archivo

"jre/lib/ext/jai_core.jar" dentro del directorio del SDK" y "Para comprobar si lo tiene

instalado compruebe que existe el archivo "jre/lib/ext/jai_imageio.jar" dentro del

directorio del SDK" y estos archivos aparecen allí.

Cualquier otra recomendación será bienvenida.

Muchas gracias.

6.5.2.4 JUMP

JUMP: Extraído de [JP 2003]

JUMP es un Sistema de Información Geográfica de carácter general orientado a la visualización y procesamiento de datos espaciales. Cuenta con las herramientas básicas de cualquier software de este tipo y ofrece una interfaz gráfica de usuario (GUI) muy intuitiva y amigable. Está licenciado bajo la licencia GPL y el código fuente está programado íntegramente en Java.

JUMP se conoce oficialmente como JUMP Unified Mapping Project y forma parte de un proyecto mayor llamado "The Jump Project". Este proyecto cuenta con otros 2 subproyectos: el JTS (JTS Topology Suite) y el JCS (JCS Conflation Suite). JUMP, en su origen, fue concebido como un framework para el desarrollo y ejecución de aplicaciones y procesamientos de datos espaciales programados a medida, es decir, una plataforma sobre la que realizar algoritmos de carácter geoespacial. Ofrece además una API completa con acceso a todos los objetos y funciones (entrada/salida, visualización, operaciones geográficas, etc). Las características de JUMP lo hacían un GIS de escritorio robusto y simple, pero además extensible dado su alto grado de modularidad y sus mecanismos para la incorporación de plugins.

Luego de la liberación de la versión JUMP 1.1.2 y de JUMP Conflation Suite en 2004 el desarrollo prácticamente se detuvo debido a la pérdida del soporte financiero. Después de la adopción de la asociación española AGILE y del proyecto francés SIGLE, un grupo de voluntarios fundó en 2004 el proyecto JPP (JUMP Pilot Project) y continuaron el desarrollo bajo el nombre de OpenJUMP.

Esta aplicación permite manejar archivos vectoriales de formato ESRI Shapefile, GML, DXF y PostGIS y archivos raster TIFF, JPG, PNG y ECW. Permite salvar vistas de rasters georeferenciados como JPG y PNG. Permite trabajar sobre algoritmos geométricos basados en Java Topology Suite. Es multilenguaje. Existe un gran conjunto de plugins, como ser conexión a Postgis, bases de datos Oracle o ArcSDE.

La documentación de OpenJUMP en cuanto a las nuevas funcionalidades es escasa, lo mismo sucede con información para los desarrolladores, ya que la misma se

encuentra bastante dispersa y el enfoque se encuentra más orientado para el desarrollo de plugins. La estructura del código fuente es bastante desordenada.

6.5.2.5 Resumen comparativo

Esta sección está basada en referencia [S 2009]

A continuación se muestra una tabla comparativa sobre las aplicaciones de escritorio vistas. Todas las aplicaciones cuentan con soporte a través de listas de mail y profesional de compañías, por lo tanto se omite la caracterización de soporte a usuarios.

Pro-grama	Foco	Nivel de usuario	Sistema operativo	Plata-forma de desarrollo	Desarrolla-do por	Licen-cia
GRASS (1982)	Análisis y visualización científica, cartografía, modelado y optimización	Experimen-tado, ..., investiga-dor	MS-Win, Linux, MacOS X	C, Shell, Tcl/Tk, Python	Institutos de investigación, Universidades, Compañías, Voluntarios alrededor del mundo	GPL
gvSIG (2003)	Visualización, edición, análisis	Novicio, ..., investiga-dor	MS-Win, Linux, MacOS X	JAVA	Compañía, Universidades, Gobierno	GPL
Map Window (1998)	Proveedor de funcionalidades GIS básicas y de interfaz gráfica, desarrollo para	Novicio, ..., investiga-dor	MS-Win	MS-Visual Studio .NET (C++, C#, VB.NET)	Universidades, Compañías, Voluntarios alrededor del mundo	MPL v1.1

	sistemas de soporte a las decisiones					
JUMP/ OpenJU MP (2002/3)	Visualización, edición, análisis	Novicio, ..., investigador	MS-Win, Linux, MacOS X	JAVA	Compañía, Gobierno, Voluntarios alrededor del mundo	GPL

Niveles de usuario: novicio (visualización), experimentado (edición, análisis simple), experto (análisis), investigador (scripting, programador).

El siguiente esquema muestra a modo de resumen las funcionalidades de las aplicaciones de escritorio en comparación con el software de la empresa ESRI ArcView9.2 un software propietario con las mismas características.

L – Lectura, E - Escritura, P - plugin

Funcionalidad		GRASS 6.3.0	gvSIG 1.1	MapWindow 4.5	OpenJum p 1.2D	ArcView 9.2
Vecto r L/E	SHP	L/E (OGR)	L/E	L/E	L/E	L/E
	GML2	L/E (OGR)	L/E	-	L/E(FME)	L
	DXF	L/E (OGR)	L/E	L(P)	L(P)/E(P)	L/E
Raster L/E	JPEG	L/E(GDAL)	L/E	L/E	L/E	L/E
	GeoTIF F	L/E(GDAL)	L/E	L/E	L	L/E
	ECW	L/E(GDAL)	L/E	L	L(P)	L
	Arc/Info GRID	L/E(GDAL)	E	L/E	E(P)	L/E
Base	PostGIS	L/E(P, limitado)	L/E	L/E(P)	L/E(P, limitado)	-

de	ArcSDE	-	L/E	-	L(P)	L
Datos L/E	Oracle	L	L/E	-	L(P)	L(OLEDB)
OGC Compliance (estándares soportados)	WMS, WFS, GML, WPS (via pyWPS)	WMS, WFS, WCS, CSW, Filter, WFS-G	WMS(P), WFS(P)	SFS, WMS, GML, WFS/T(P)	WMS, WCS, SFS, GML	
Mapeo temático (gráficas, símbolos, valores personalizados, etiquetas, rangos, densidad de puntos)	Si	Si(simple)	- (solo etiquetas)	Si(P)	Si	
Impresión	Si	Si	Si	Si	Si	
API Desarrollo	Si	-	Si	Si	-	
Funcionalidad de scripting	Bash, Python, Perl	Jython	C#, VB.NET	Bean shell, Jython(P)	-	
Transformación y proyección de coordenadas	Si	Si(P)	Si	Si(P)	Si	
Creación y edición	Si	Si	Si	Si	Si	
Soporte GPS	Si	-	Si	Si(P)	Si	
Creación de topologías (aristas, nodos, cadenas y	Si	-	- (sólo API) creación de TIN)	Si(limitado)	-	

topología de polígonos)					
Creación y edición avanzada (offset, generalización de líneas, recortes, rotación, intersección, etc)	Si	Si(P)	Si(excepto generalización)	Si(excepto offset)	Si(excepto offset, generalización de líneas)
Análisis de polígonos de Thiessen, análisis de grillas, generación de contornos y líneas de flujo	Si	Si(P)	Si(comparación de contornos y líneas de flujo)	(solo polígonos de Thiessen)	-
Creación de vistas 3D y terrenos (TIN, DIM, sombreado, drapeado de superficie)	Si	Si(P)	Si(3D plugin, TIN solo para API, hillshades)	Si(P)	(solo display de superficie)
ViewShed y análisis de terreno (línea de visión, pendiente, aspecto, gradiente)	Si	Si(P)	Si(análisis de cuenca)	-	-

6.5.3 Bibliotecas

6.5.3.1 GeoTools

GeoTools: Extraído de [AV 2006] y [GT 2009]

Es una biblioteca SIG de código libre que permite desarrollar soluciones adaptadas a los estándares. Proporciona una implementación de las especificaciones del OGC según van apareciendo.

Está escrito en el lenguaje de programación Java y se encuentra actualmente en un desarrollo activo al nutrirse de una comunidad de usuarios muy dinámica. Su diseño y concepción modular hace que numerosas implementaciones de software libre en el ámbito de los SIG hagan uso de los desarrollos de GeoTools.

El código liberado de GeoTools está bajo la licencia GNU/ LPGL.

El trabajo en la biblioteca GeoTools comenzó en el año 1996 en la Universidad de Leeds como parte de un proyecto de maestría, para poder visualizar los resultados de una Máquina de Análisis Geográfico (GAM) en la Web. Poco tiempo después se desplegó el primer mapa. En conjunto con el departamento geográfico de la universidad, se lanzó la investigación PPGIS (SIG de Participación Pública) en el cual mediante el uso de la biblioteca de GeoTools se creó un mapa on-line de la ciudad, permitiendo a los usuarios resolver tareas de planificación. La biblioteca fue desarrollada rápidamente en consecuencia a las necesidades de un segundo proyecto, consiguiendo un conjunto de herramientas basadas en la API Java Applet, surgiendo así el GeoTools-Lite.

Soporta la implementación OGC Grid Coverage. Soporta sistemas de coordenadas y transformaciones. La simbología sigue la especificación de OGC Styled Layer Description (SLD). Atributos y filtros espaciales siguen la especificación de OGC Filter Encoding. Soporta modelado de grafos y redes. Java Topology Suite (JTS) – con soporte para la especificación OGC Simple Features Specification – usado para el modelado geométrico de vectores.

Soporta varios formatos vectoriales como: ESRI Shapefile, OGC GML, OGC WFS, PostGIS, Oracle Spatial, ArcSDE, MySQL. En cuanto a raster soporta los formatos:

OGC WMS, ArcGrid, GeoTIFF, imágenes georeferenciadas con un archivo de mundo.

La documentación para desarrolladores es pobre.

6.5.3.2 FDO

FDO – Extraído de [FDO 2007]

FDO (Feature Data Objects) es una API para manipular, definir y analizar información GEO espacial sin importar donde esta almacenada dicha información. FDO utiliza un modelo basado en los proveedores para así soportar varias fuentes de información GEO espacial. Un proveedor es una implementación específica de la API de FDO que provee acceso a la información en un tipo de almacenamiento de datos en particular.

La empresa Autodesk necesitaba una tecnología que pudiera proveer una forma de acceso a distintas fuentes de datos geoespaciales mediante un mecanismo común. Las fuentes de datos incluían una variedad de bases de datos geoespaciales y archivos con diversos formatos. Por lo tanto debía ser un producto flexible para poder cumplir con los requisitos de muchas aplicaciones de cliente.

Los requisitos iniciales incluían los siguientes puntos:

- Proveer un modelo de datos geoespaciales orientados a objetos, que sea genérica e independiente con respecto a cualquier implementación nativa.
- Soportar como mínimo la especificación de tipos geométricos establecida por la OGC Simple Feature.
- Definir un modelo lógico que se corresponda con las implementaciones físicas más comunes, como son Oracle Spatial y archivos SHP.
- Definir un proveedor FDO como una implementación de la FDO API para una fuente de datos particular. Cada fuente de datos deberá tener su proveedor FDO correspondiente.
- Definir cualidades de la API permitiendo a los clientes determinar cualidades específicas para un proveedor particular.
- Permitir nuevos comandos, modificaciones de esquemas sobre la API genérica. Además de permitir agregar comandos personalizados a cualquier proveedor particular.

La primer versión de FDO salió al mercado junto al producto Autodesk Map 3D 2005, a fines del 2004. Se incluían proveedores para Oracle y SDF. La versión 2.0

agregó un proveedor de ArcSDE y la versión 3.0 sumó los proveedores para MySQL, SQL Server, ODBC, SHP, Raster, OGC WFS y OGC WMS.

La liberación al mercado de FDO con código libre fue junto a la salida de Map Guide con código libre en el año 2006. Incluye proveedores para SDF, SHP, MySQL, ArcSDE, ODBC, OGC WFS y OGC WMS.

Contiene algunos textos que explican los conceptos básicos de la interacción con los distintos proveedores, acceso y creación de tablas y esquemas.

6.5.3.3 Resumen comparativo

La Tabla 6.5.3.3.1 se muestra un par de tablas comparativas sobre las bibliotecas estudiadas.

Librería	Foco	Sistema operativo	Plataforma de desarrollo	Desarrollado por	Licencia
GeoTools (1996)	Visualización, edición, análisis, WEB	MS-Win, Linux, MacOS X	JAVA	Universidades, Voluntarios alrededor del mundo	LGPL
FDO (2004)	Brindar acceso uniforme a distintas fuentes de datos geográficos	MS-Win, Linux	C++	Compañía, Universidades, Voluntarios alrededor del mundo	LGPL

Tabla 6.5.3.3.1

La Tabla 6.5.3.3.2 muestra a modo de resumen las funcionalidades de las bibliotecas en comparación con el software de la empresa ESRI MapObject2.4 un software propietario con las mismas características.

L – Lectura, E - Escritura

Funcionalidades		GeoTools 2.5.X	FDO 3.2	MapObject 2.4
Vector	SHP	L/E	L/E	L/E
L/E	GML2	L	L/E	-

	DXF	-	L/E	L/E
Raster	JPEG	L/E	L	L/E
L/E	GeoTIFF	L	L	L/E
	ECW	-	L	-
	ArcInfo GRID	L/E	L	L/E
Base de	PostGIS	L	-	-
Datos	ArcSDE	L	L/E	L/E
L/E	Oracle	L	L/E	L/E
OGC Compliance (estándares soportados)		WMS(L), WFS, GML(L)	WMS, WFS	-
Mapeo temático (gráficas, símbolos, valores personalizados, etiquetas, rangos, densidad de puntos)		Si	-	Si
Transformación y proyección de coordenadas		Si	-	Si
Creación y edición		Si	Si	Si
Creación de topologías (aristas, nodos, cadenas y topologías de polígonos)		-	-	Si
Creación y edición avanzada (offset, generalización de líneas, recortes, intersección, etc)		Si	-	Si

Tabla 6.5.3.3.2

6.5.4 Evaluación del Software analizado

Se evaluaron distintas características para poder seleccionar una de las herramientas. Lo primero y más importante era que cumpliera con lo mínimo necesario para poder desarrollar la aplicación. Con respecto a esto, las necesidades básicas como se comentó con anterioridad (2.2 Requisitos No Funcionales), refieren a un completo manejo del formato ESRI Shapefile, creación, consulta y visualización de capas, agregar y eliminar elementos de las capas. Todas las herramientas cumplían con este cometido por lo tanto se decidió evaluar los siguientes criterios: facilidad de instalación, armado de ambiente de desarrollo y plataforma de trabajo. En base a estos criterios se construyó la Tabla 6.5.4.1.

Programa	Tiempo de aprendizaje	Lenguaje	Creación de ambiente de desarrollo
MapWindow	Rápido	.Net (C#, C++, VB)	Fácil
GRASS	Medio	C, Tcl-Tk, shell	-
GvSIG	Rápido/Medio	Java	Media
JUMP	Rápido/Medio	Java	Media
GeoTools	Rápido/Medio	Java	Media
FDO	Medio	C++	Fácil

Tabla 6.5.4.1

La medida del tiempo de aprendizaje la tomamos a partir de las funcionalidades ofrecidas por el producto, documentación de las funcionalidades, conocimientos y productividad del lenguaje de programación para desarrollar.

Se debía completar una primera versión de la aplicación para el mes de diciembre de 2008. Esta versión debía ser lo suficientemente estable como para poder trabajar sin problemas sobre ella.

Sin lugar a duda la herramienta SIG que presentaba más funcionalidades era GRASS, el inconveniente que se encontró fue que el costo de aprender a desarrollar en

esta herramienta implicaría más tiempo del que se tenía disponible. Esto se debió principalmente, al desconocimiento de herramientas para construir la interfaz gráfica utilizando el lenguaje C, por desconocer el lenguaje Tcl/Tk o Python.

Las herramientas JAVA resultaban atractivas sobre todo por ser multi plataforma al igual que GRASS, pero fue demasiado complicado establecer el ambiente de desarrollo y la documentación de desarrollador no era buena.

FDO es una buena librería para el acceso a la información geográfica en distintos repositorios dada su capacidad de abstracción de los proveedores, pero se limita a esa importante funcionalidad lo cual no nos resultaba suficiente para poder desarrollar la aplicación, por lo tanto no fue considerada para la elección final.

MapWindow es una herramienta que no ofrece tantas funcionalidades como GRASS o gvSIG, pero tiene todas las funcionalidades que se necesitaban. Fue la primera herramienta para la que se logró establecer un buen ambiente de desarrollo como consecuencia de su fácil de instalación. Tiene muy buena documentación de desarrollador con una API actualizada.

Finalmente y para concluir la decisión fue utilizar MapWindow. El único punto en contra de MapWindow es que sólo corre en MS-Win, si bien en los requisitos funcionales no se especifica una plataforma en concreto, no logramos una aplicación multi plataforma.

7. Anexo. Entorno de Desarrollo

Una vez determinado la herramienta SIG a utilizar, se desprendieron las alternativas para entornos de desarrollo del software. Estos se vieron acotados a la familia de Visual Studio .NET. Los lenguajes podían ser C++, Visual Basic o C#. La primera alternativa fue descartada debido a la baja productividad de líneas de código por hombre, mencionadas en proyectos de grado en años anteriores [MF 2003]. Nuestro conocimiento sobre cualquiera de las otras dos opciones era muy limitado, así que luego de una búsqueda reducida en tiempo, se notó que ambos lenguajes eran similares en sus nociones básicas, inclinando la decisión hacia C#. Otra consideración fue la de intentar utilizar los últimos avances, en cuanto a entornos de desarrollo, ya que se creyó conveniente para mantener la aplicación lo más actualizada posible. Debido a esto igoR-tp fue desarrollado en Visual Studio .NET 2008, con lenguaje C#.

Aprovechando el uso de la última tecnología se decidió emplear Windows Presentation Foundation (WPF) para el desarrollo de las interfaces gráficas de la herramienta. Esta decisión se basó en las dificultades que existe hoy día en el desarrollo de interfaces gráficas, debido a que los métodos de colaboración entre diseñadores y desarrolladores son ineficientes e insuficientes. Si bien el proyecto no contaría con un grupo de programadores y otro de diseñadores, siempre es positivo mantener diferenciada la presentación gráfica, de la lógica y los otros componentes de la aplicación.

Otros obstáculos presentes entre el diseño de la interfaz gráfica y la programación son:

- Comunicación de una sola vía: diseñador hacia desarrollador. La comunicación entre el diseñador y el desarrollador está basada en los prototipos que el primero genera.
- Dificultades para introducir cambios en el diseño luego de que el desarrollador ha comenzado a implementar los prototipos.

Generalmente, el resultado gráfico se encuentra por debajo de las expectativas.

La ausencia de buenas herramientas de colaboración entre diseñadores y desarrolladores provoca fricción entre ambos equipos.

A su vez han surgido nuevas necesidades en cuanto a la presentación gráfica. Ellas se deben a:

- La experiencia de uso de las herramientas tiene igual importancia que su eficacia. Ya no es suficiente que el usuario pueda cumplir con sus tareas sino que se debe brindar una buena experiencia de uso.
- El uso masivo de aplicaciones en Internet ha aumentado el nivel de exigencia de los usuarios. Los usuarios se han acostumbrado el uso de aplicaciones en Internet y esperan encontrar en las aplicaciones de escritorio las mismas prestaciones y presentaciones que las encontradas en la Web.

Debido a los obstáculos y necesidades de los propios usuarios fue como surgió WPF [AS 2008], que es un marco de presentación de interfaces de usuario que integra gráficos 2D Y 3D, efectos, animación, audio, video y despliegue de documentos.

Las características principales son que soporta enlace con fuentes de datos y permite el uso de estilos y plantillas.

Algunos de sus beneficios son: permitir definir la interfaz gráfica en forma declarativa, posibilita distribuir las aplicaciones como 'Standalone' o vía Web, explotar el poder de procesamiento brindado por los 'GPU'.

Otra característica del WPF es el uso de lenguaje XAML (Extensible Application Markup Language). Éste se ha posicionado como un lenguaje de diseño de interfaces gráficas. Permite definir la interfaz gráfica en forma declarativa, separada del comportamiento. Está basado en XML.

En WPF, las aplicaciones son encapsuladas en la clase Application. Esta provee la infraestructura común para toda la aplicación, administrando las ventanas y proveyendo las propiedades y recursos comunes.

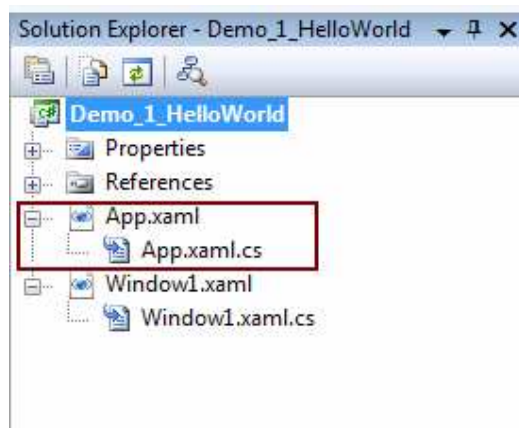


Figura 7.1. Explorador de soluciones de Visual Studio .NET donde se ve la clase Application en XAML.

Microsoft provee herramientas para desarrolladores y diseñadores utilizando XAML como medio de integración.

Las características que se presentaron acerca de esta nueva tecnología no fueron explotadas en su máximo alcance, pero si se intentó aprovechar los conceptos básicos de la misma lo mejor posible. Por otra parte cabe remarcar que WPF brinda infinitas posibilidades de mejoras en la interfaz gráfica de iGoR-tp, sin incidir en lo más mínimo

en la lógica u otras partes de la aplicación, lo cual se traduce en una gran ventaja para los trabajos a futuro sobre la apariencia de la misma.

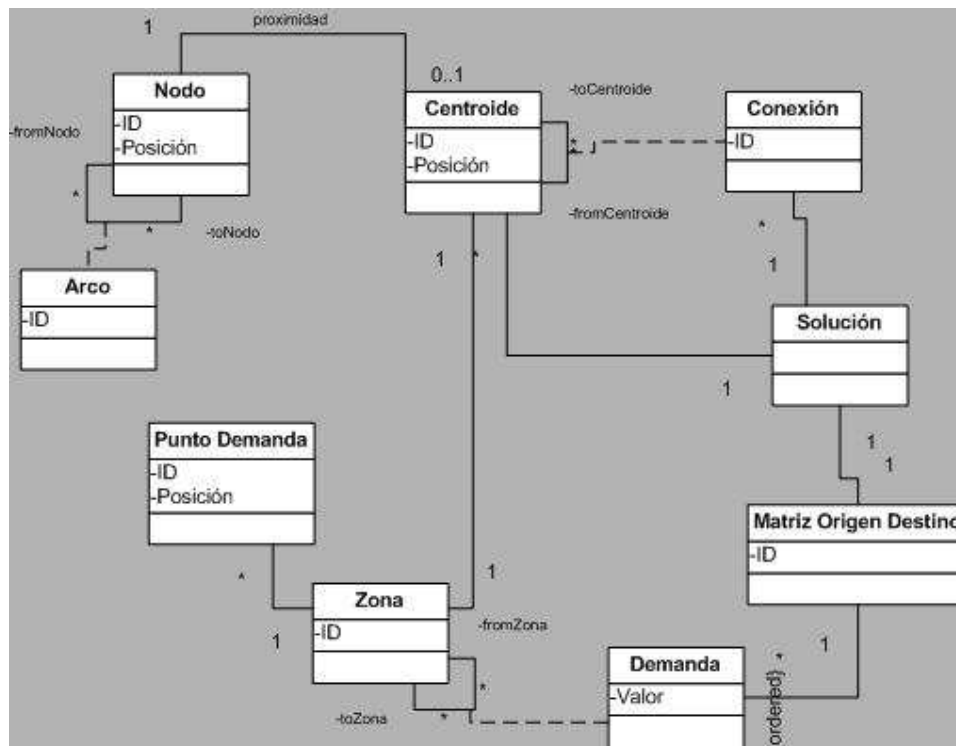
8. Anexo. Análisis y Diseño

Este anexo describe las etapas de análisis y diseño del problema. En el análisis se enfrentó el relevamiento de requisitos, desembocando en el modelo de dominio, junto a los casos de uso. Una vez modelado el problema se edificó el diseño de la solución del cual se desprendió la arquitectura de la herramienta a implementar posteriormente.

8.1 Modelo de Dominio

8.1.1 Modelo de Dominio - Construcción del Caso de Estudio

8.1.1.1 Diagrama:



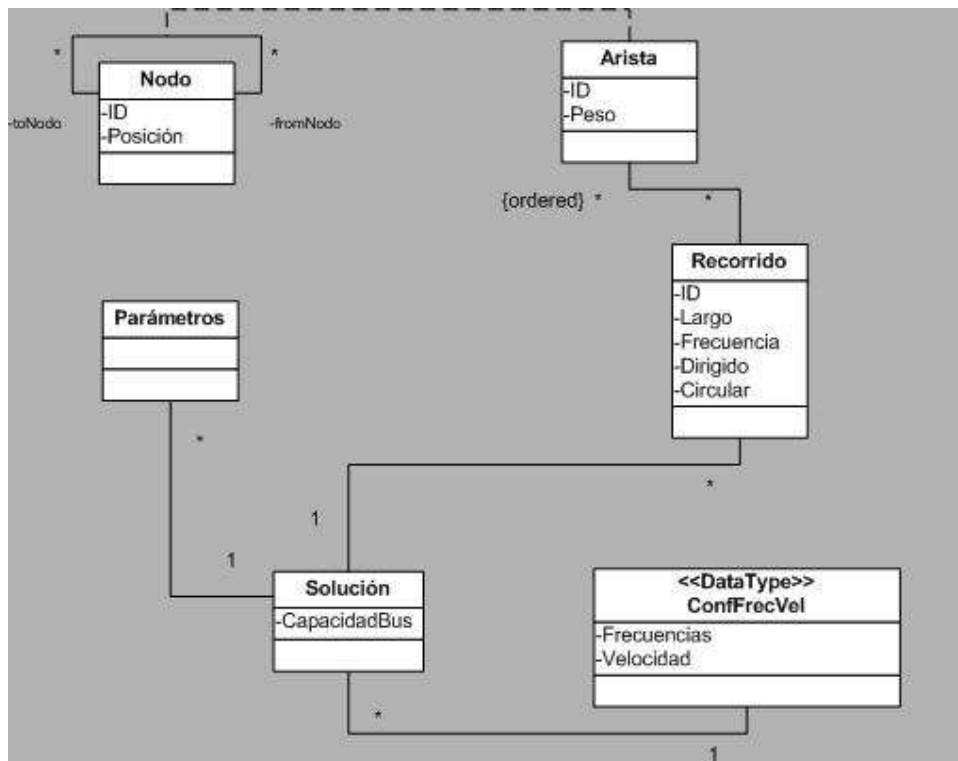
8.1.1.2 Restricciones:

- Ninguna conexión tiene el mismo centroide en fromCentroide y toCentroide.
- Ninguna demanda tiene la misma zona en fromZona y toZona.
- No hay dos aristas con el mismo par de vértices asociados.
- No hay dos conexiones con el mismo par de centroides asociados.

- No hay dos demandas con las mismas zonas en toZona y fromZona.
- El peso de las conexiones es igual a la suma de los pesos de las aristas necesarias para ir desde el vértice más próximo asociado al centroide fromCentroide, hasta el vértice más próximo asociado al centroide toCentroide por un camino mínimo.
- La posición de los centroides es calculada a partir de los datos de la zona.
- El vértice fromVert de la primera arista es igual al vértice toVert de la última arista de una zona.

8.1.2 Modelo de Dominio - Manipulación del Caso de Estudio

8.1.2.1 Diagrama:



8.1.2.2 Restricciones:

- Ninguna arista tiene el mismo vértice en fromNodo y toNodo.
- No hay dos aristas que tengan los dos mismos vértices asociados.
- El largo de los recorridos se calcula como la suma de los pesos de las aristas asociadas.

8.2 *Análisis de requisitos*

8.2.1 Resumen

El sistema a construir es una herramienta con funcionalidades de interfaz gráfica y manipulación de datos geográficos que será utilizada para la planificación de recorridos para transporte público.

Este sistema se puede dividir en dos subsistemas. El primer subsistema se encarga de construir los datos de un caso de estudio. Un caso de estudio es el conjunto de información necesaria para que el sistema pueda llevar a cabo todas sus funcionalidades. Puede incluir datos topológicos y datos de demanda (en las siguientes secciones de este anexo se explican tanto la información de un caso de estudio como las funcionalidades del sistema). Este recibe la red vial de una ciudad y le permite al usuario construir mediante el mecanismo de zonificación un grafo con mayor nivel de agregación que la red vial. El segundo subsistema se encarga de interactuar con los algoritmos que planifican los recorridos, y desplegarle al usuario estas soluciones, además de permitirle evaluar las soluciones que crea conveniente.

8.2.2 Descripción

El sistema constará de 2 módulos principales:

- Módulo de Construcción del Caso de Estudio.
- Módulo de Manipulación del Caso de Estudio.

8.2.2.1 Módulo de Construcción del Caso de Estudio

8.2.2.1.1 Descripción

Dado un conjunto de datos de entrada, que involucra principalmente una red y datos de demanda de viajes, este módulo permite al usuario definir un conjunto de zonas. Las zonas deben cumplir tres requisitos: contigüidad espacial, compacidad espacial y compatibilidad con líneas existentes. La contigüidad espacial básicamente restringe las zonas a polígonos simples, es decir que no contienen islas ni agujeros, además no se permiten zonas que encierren a otras zonas. La compacidad espacial y compatibilidad con líneas existentes asegura que las zonas sean compatibles con la topología (por ejemplo las aristas de las zonas coinciden con las calles). La propiedad de

compacidad es deseable pero no será exigida a los usuarios. Para esto el módulo debe desplegar en pantalla información suficiente para que el usuario pueda llevar a cabo este cometido (la definición de las zonas). Durante el ingreso de zonas el módulo debe calcular y desplegar, centroides (uno por zona). Posteriormente se podrán realizar conexiones entre centroides, las cuales tendrán pesos asociados. La construcción de las conexiones entre centroides se realiza manualmente. Otra funcionalidad será la de calcular la demanda entre centroides. Además, el módulo debe poder persistir la construcción del caso de estudio descrita brevemente en las líneas anteriores, de forma que pueda ser retomada en otro momento.

8.2.2.1.2 Entrada

El conjunto de datos de entrada que recibe este módulo se puede dividir en dos subconjuntos:

- datos topológicos: los datos topológicos deben contener la información necesaria para describir la red del caso de estudio. En nuestro sistema, dicha red se representará mediante una capa de datos geográficos. Específicamente, una capa de polilíneas que llamaremos “capa de aristas” que debe contener al menos dos atributos que indiquen sus puntos extremos utilizando los cuales deben estar identificados por un entero. La capa debe encontrarse en formato ESRI Shapefile.

- datos de demanda: los datos de demanda deben contener la información para describir la demanda sobre la red. Esto quiere decir, que debe ser posible determinar la demanda entre los centroides de un caso de estudio. Esto implica datos numéricos y datos geográficos. Los datos numéricos estarán almacenados en una base de datos Access. Estos datos se relacionan con una capa de puntos (datos geográficos), que llamaremos “capa de puntos demanda”, cada punto debe tener un identificador el cual será utilizado para relacionarlos con los datos numéricos.

8.2.2.1.3 Salida

Como se mencionó en la descripción, este módulo debe ser capaz de persistir el caso de estudio, el cual servirá de entrada al módulo de manipulación.

8.2.2.2 Módulo de Manipulación del Caso de Estudio

8.2.2.2.1 Descripción

Dado un conjunto de datos de entrada, que como elemento principal contiene un grafo, este módulo debe ser capaz de invocar una serie de algoritmos que se aplicarán sobre el grafo de entrada y permitirá desplegar la información resultado de la aplicación de los algoritmos. Esta información consiste en una serie de recorridos sobre el grafo. Además tiene que poder interactuar con el usuario, el cual puede realizar modificaciones a los recorridos desplegados con el cometido de reevaluar el nuevo caso, mediante la invocación a los algoritmos.

8.2.2.2.2 Entrada

El conjunto de datos de entrada de este módulo se puede dividir en cuatro subconjuntos:

- Datos topológicos: los datos topológicos hacen referencia al grafo mencionado en la descripción. Este grafo contiene la información acerca de sus vértices y sus aristas. Como mínimo de información debe contener el peso de las aristas.
- Datos de demanda: este módulo solo necesita los datos de demanda para la invocación a los algoritmos. Los datos de demanda están expresados mediante una matriz origen destino.
- Característica de la flota: los datos sobre la característica de la flota son necesarios para la ejecución de los algoritmos. Incluyen datos como, capacidad de pasajeros sentados y velocidad media.
- Parámetros de diseño: los parámetros de diseño son necesarios para la ejecución de los algoritmos. Incluyen datos como, proporciones mínimas de demanda cubierta, frecuencias mínimas y máximas, factor de carga máximo de los buses. Los parámetros de diseño dependen del algoritmo que se quiera ejecutar.

El módulo debe poder cargar algún tipo de capa como fondo para que el usuario tenga, si lo desea, alguna referencia extra.

8.2.2.2.3 Salida

El módulo debe poder persistir las soluciones de los algoritmos y las editadas por el usuario.

8.2.3 Usuarios

La división de los requisitos en dos módulos requiere que el usuario realice distintos esfuerzos en el manejo de estos. El módulo de construcción del caso de estudio demanda un mayor esfuerzo en tiempo. Aunque la herramienta automatiza gran parte del proceso, el ingreso de las zonas es principalmente manual. Por otra parte, el módulo de manipulación del caso de estudio, requiere un mayor esfuerzo en el conocimiento de los algoritmos de optimización. Aunque la herramienta proponga valores para los parámetros de los algoritmos, es necesario que el usuario sepa que representan estos parámetros.

8.2.4 Requisitos Funcionales

A continuación se describen los requisitos funcionales divididos según el módulo.

8.2.4.1 Requisitos Funcionales - Módulo de Construcción del Caso de Estudio

8.2.4.1.1 Nuevo Proyecto

El sistema debe permitir crear un nuevo proyecto con el nombre y ruta que el usuario desee.

8.2.4.1.2 Abrir Proyecto

El sistema debe permitir abrir un proyecto previamente creado.

8.2.4.1.3 Cerrar Proyecto

El sistema debe permitir cerrar el proyecto.

8.2.4.1.4 Guardar Proyecto

El sistema debe permitir guardar el proyecto.

8.2.4.1.5 Guardar Proyecto Como

El sistema debe permitir guardar el proyecto “activo” con el nombre y ruta que el usuario desee.

8.2.4.1.6 Salir

El sistema debe permitir salir del mismo.

8.2.4.1.7 Agregar Fondo

El sistema debe permitir agregar nuevos fondos al mapa.

8.2.4.1.8 Quitar Fondo

El sistema debe permitir eliminar fondos del mapa.

8.2.4.1.9 Full Extent

El sistema debe permitir visualizar todo el mapa.

8.2.4.1.10 Zoom In

El sistema debe permitir acercar el mapa tomando como centro de visualización una posición seleccionada del este.

8.2.4.1.11 Zoom Out

El sistema debe permitir alejar el mapa tomando como centro de visualización una posición seleccionada de este.

8.2.4.1.12 Pan

El sistema debe permitir al usuario mover el mapa en distintas direcciones, para visualizar las distintas partes de este.

8.2.4.1.13 Información de capas

El sistema debe permitir al usuario obtener información acerca de las capas que fueron entrada del proyecto, o son utilizadas como fondo.

8.2.4.1.14 Agregar Zona

El sistema debe permitir crear nuevas zonas en el proyecto activo.

8.2.4.1.15 Eliminar Zona

El sistema debe permitir eliminar zonas en el proyecto activo.

8.2.4.1.16 Agregar Conexión

El sistema debe permitir crear nuevas conexiones entre pares de zonas del proyecto activo.

8.2.4.1.17 Eliminar Conexión

El sistema debe permitir eliminar conexiones en el proyecto activo.

8.2.4.1.18 Generar Matriz Origen Destino

El sistema debe permitir a partir de la información desprendida de las zonas y conexiones construidas y los datos sobre la demanda de una ciudad y un horizonte horario, generar una matriz de demanda que englobe esa información. Que contenga para cada par de zonas las demandas que existen.

8.2.4.1.19 Activar Notas

El sistema debe permitir activar notas sobre el mapa para visualizarlas en el proyecto activo. La nota es información sobre las capas de entrada del proyecto o de aquellas capas que hayan sido agregadas como fondo.

8.2.4.1.20 Desactivar Notas

El sistema debe permitir desactivar las notas sobre el mapa en el proyecto activo.

8.2.4.1.21 Personalizar Notas

El sistema debe permitir personalizar la información a desplegar en las notas sobre el mapa del proyecto activo.

8.2.4.1.22 Información matriz Origen Destino

El sistema debe permitir observar los datos de la matriz origen destino generada.

8.2.4.1.23 Opciones.

El sistema debe permitir distintas posibles visualizaciones sobre el mapa. Como ser el camino mínimo sobre la red de aristas, la conectividad de la red de aristas y los puntos de demanda sin zonificar.

8.2.4.2 Requisitos Funcionales - Módulo de Manipulación del Caso de Estudio

8.2.4.2.1 Nuevo Proyecto

El sistema debe permitir crear un nuevo proyecto con el nombre y ruta que el usuario desee.

8.2.4.2.2 Abrir Proyecto

El sistema debe permitir abrir un proyecto previamente creado.

8.2.4.2.3 Cerrar Proyecto

El sistema debe permitir cerrar el proyecto.

8.2.4.2.4 Guardar Proyecto

El sistema debe permitir guardar el proyecto.

8.2.4.2.5 Guardar Proyecto Como

El sistema debe permitir guardar el proyecto con el nombre y ruta que el usuario desee.

8.2.4.2.6 Salir

El sistema debe permitir salir del mismo.

8.2.4.2.7 Agregar Fondo

El sistema debe permitir agregar nuevos fondos al mapa.

8.2.4.2.8 Quitar Fondo

El sistema debe permitir eliminar fondos del mapa.

8.2.4.2.9 Nuevo Recorrido

El sistema debe permitir crear nuevos recorridos en nuevas soluciones o en soluciones ya existentes.

8.2.4.2.10 Eliminar Recorrido

El sistema debe permitir eliminar recorridos.

8.2.4.2.11 Agregar Tramo

El sistema debe permitir agregar nuevos tramos al recorrido activo.

8.2.4.2.12 Eliminar Tramo

El sistema debe permitir eliminar tramos del recorrido activo.

8.2.4.2.13 Abrir Solución

El sistema debe permitir abrir soluciones ya existentes y cargarlas en el mapa. Recordemos que una solución es un conjunto de recorridos con sus respectivas frecuencias.

8.2.4.2.14 Cerrar Solución

El sistema debe permitir cerrar la solución abierta.

8.2.4.2.15 Full Extent

El sistema debe permitir visualizar todo el grafo.

8.2.4.2.16 Zoom In

El sistema debe permitir acercar el grafo tomando como centro de visualización una posición seleccionada de este.

8.2.4.2.17 Zoom Out

El sistema debe permitir alejar el grafo tomando como centro de visualización una posición seleccionada de este.

8.2.4.2.18 Pan

El sistema debe permitir al usuario mover el grafo en distintas direcciones, para visualizar las distintas partes de este.

8.2.4.2.19 Generar Recorridos

El sistema debe permitir crear recorridos en una solución generada por los algoritmos.

8.2.4.2.20 Evaluar Recorridos

El sistema debe permitir evaluar recorridos sobre el grafo utilizando los algoritmos.

8.2.4.2.21 Graficar evaluación

El sistema debe permitir desplegar la evaluación de una solución mediante una gráfica. La misma será el histograma de perfil de carga.

8.2.4.2.22 Activar Notas

El sistema debe permitir activar notas sobre el mapa para visualizarlas en el proyecto activo. La nota es información sobre las capas de entrada del proyecto o de aquellas capas que hayan sido agregadas como fondo.

8.2.4.2.23 Desactivar Notas

El sistema debe permitir desactivar las notas sobre el mapa en el proyecto activo.

8.2.4.2.24 Personalizar Notas

El sistema debe permitir personalizar la información a desplegar en las notas sobre el mapa del proyecto activo.

8.2.4.2.25 Información de capas

El sistema debe permitir al usuario obtener información acerca de las capas que son utilizadas como fondo en el proyecto activo.

8.2.4.2.26 Información de Directos y Traspardos

El sistema debe permitir obtener información acerca de los recorridos directos y traspardos entre dos centroides.

8.2.4.2.27 Crear configuración de frecuencias y capacidad.

El sistema debe permitir crear un conjunto de frecuencias y capacidades aplicables a cualquier solución.

8.2.4.2.28 Eliminar configuración de frecuencias y capacidad.

El sistema debe permitir eliminar un conjunto de frecuencias y capacidades aplicables a cualquier solución.

8.2.4.2.29 Vista recorridos.

El sistema debe permitir ver recorridos en el mapa diferenciados de acuerdo a la frecuencia y a la dirección como atributos visuales.

8.2.5 Requisitos no Funcionales

8.2.5.1 Utilidad

La herramienta debe ser fácil de usar. Debe ser altamente disponible, es decir que todos los elementos para su correcto funcionamiento deben ser sencillamente accedidos por el usuario. Su instalación debe ser sencilla.

8.2.5.2 No propietaria

La herramienta debe utilizar algún sistema de información geográfica de tipo no propietaria. El código no tiene que ser necesariamente libre, pero si la obtención del sistema que manipula la información geográfica.

8.2.5.3 Documentación

Se debe incluir una completa documentación técnica y de usuario de la herramienta.

8.2.6 Interfaces

8.2.6.1 Interfaces de usuario

El sistema contará con dos interfaces gráficas bien definidas y totalmente independientes. Una para el módulo de preparación del caso de estudio y la otra para el módulo de manipulación del caso de estudio.

8.2.6.2 Interfaces con software

El sistema, en especial el módulo de manipulación del caso de estudio debe interactuar con los algoritmos que resuelven los recorridos.

En los siguientes párrafos se describen los casos de uso con sus respectivos actores y escenarios.

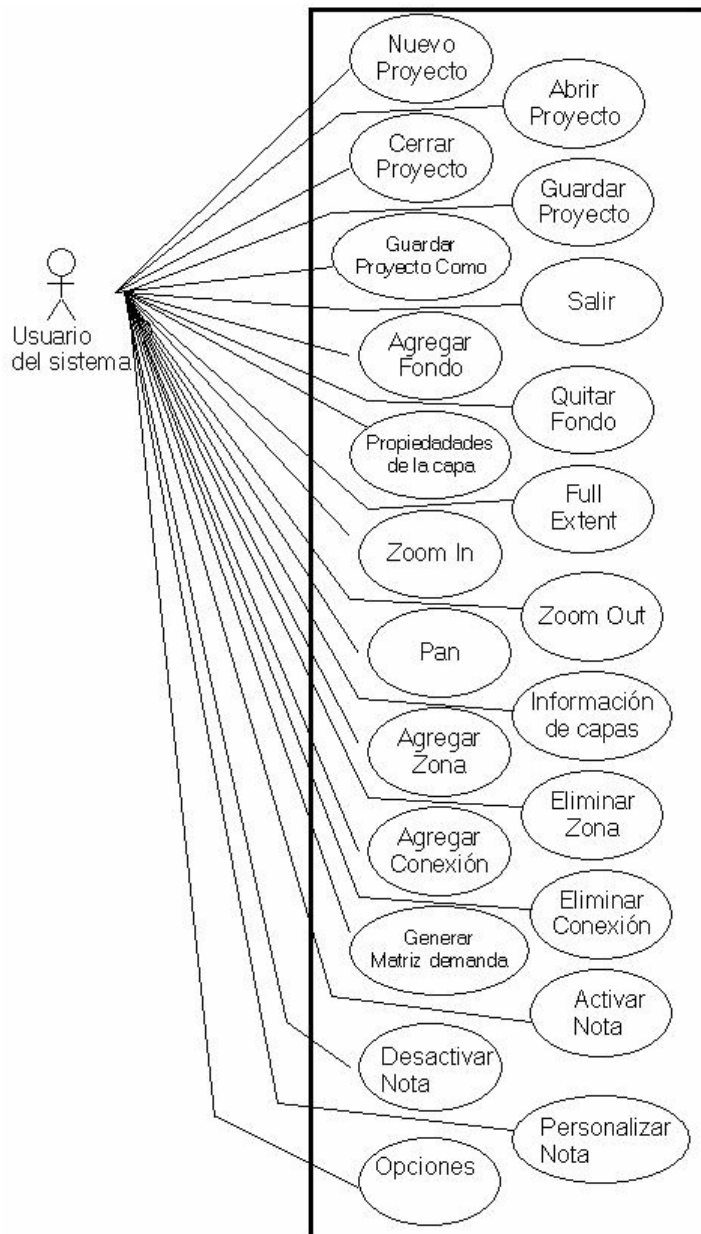
8.3 Casos de Uso

8.3.1 Usuario del Sistema

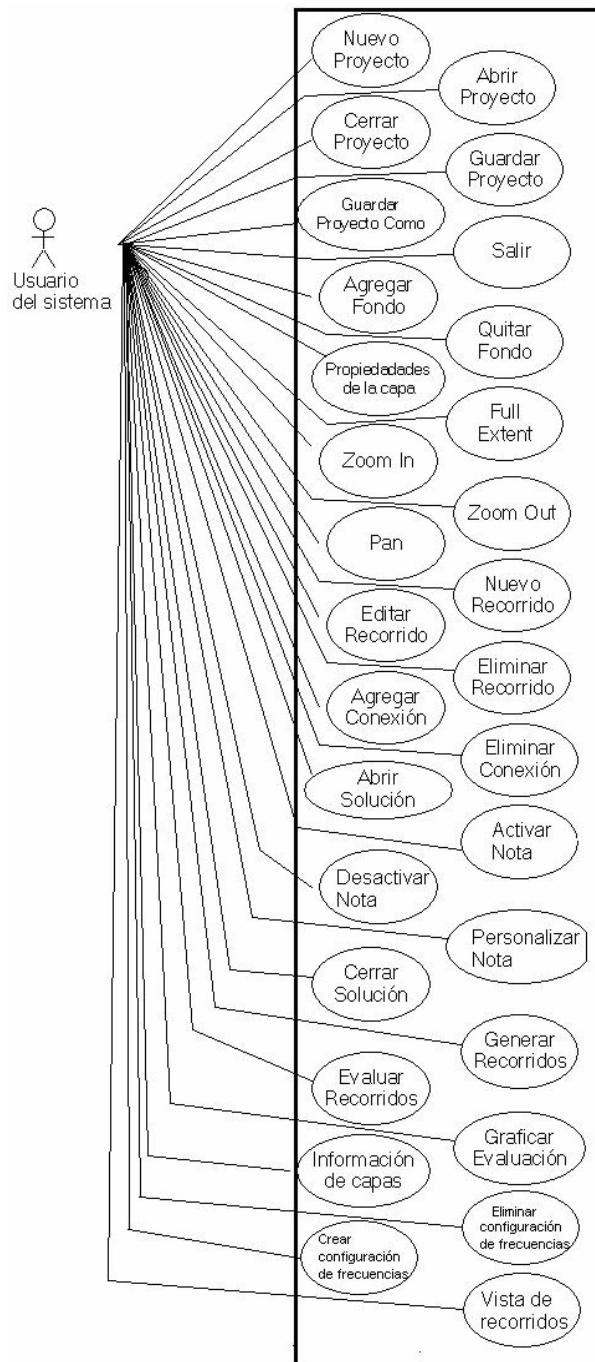
Los usuarios del sistema son personas conocedoras del dominio. No necesariamente requieren los mismos conocimientos los usuarios que interactúan con el módulo de preparación que los que interactúan con el módulo de manipulación. Los usuarios del módulo de preparación necesitan un conocimiento básico de sistemas geográficos. Los usuarios del módulo de manipulación necesitan conocimientos acerca de algoritmos de optimización. De aquí desprendemos dos tipos de usuario, el diseñador de la red, capaz de manipular (uso y desarrollo) los algoritmos de optimización; y el planificador, aquellos experto para el caso de estudio por su pericia y conocimiento de la realidad pero no familiarizados con el uso de los algoritmos.

Diagramas de Caso de Uso

8.3.1.1 Diagrama de Casos de Uso - Módulo de Construcción del Caso de Estudio



8.3.1.2 Diagrama de Casos de Uso - Módulo de Manipulación del Caso de Estudio



8.3.2 Descripción de los Casos de Uso

8.3.2.1 Casos de Uso - Módulo de Construcción del Caso de Estudio

8.3.2.1.1 Nuevo Proyecto

Breve Descripción: El usuario crea un nuevo proyecto con un nombre y ruta.

Precondiciones: -.

Poscondiciones: Se creó un proyecto con el nombre ingresado en la ruta ingresada.

Flujo Principal:

- 1- El usuario selecciona la opción nuevo proyecto.
- 2- El sistema le pide nombre y ruta.
- 3- El usuario llena los datos y acepta.
- 4- El sistema verifica que los datos sean correctos.
- 5- El sistema le pide una capa de puntos de demanda y el campo identificador de la misma
- 6- El usuario completa los datos y acepta
- 7- El sistema le pide una capa de aristas.
- 8- El usuario ingresa la ubicación de la capa, elige calcular la red de aristas y acepta.
- 9- El sistema pide se ingrese la velocidad media de la flota.
- 10- El usuario completa la información y finaliza.
- 11- El sistema crea el proyecto con los datos ingresados.

Flujos Alternativos:

Cancelación.

- 3B – El usuario cancela la operación.
- 6B – El usuario cancela la operación.
- 8B – El usuario cancela la operación.
- 10B – El usuario cancela la operación.

Datos incompletos.

- 3C – El usuario no llena todos los datos y acepta.

3C1 – El sistema le indica los campos obligatorios con un mensaje.

3C2 – Vuelve a 3.

6C – El usuario no llena todos los datos y acepta.

6C1 – El sistema le indica los campos obligatorios con un mensaje.

6C2 – Vuelve a 6.

8C – El usuario no llena todos los datos y acepta.

8C1 – El sistema le indica los campos obligatorios con un mensaje.

8C2 – Vuelve a 8

Datos incorrectos.

5D – El sistema emite el mensaje “Ya existe un proyecto con el mismo nombre”.

5D1 – Vuelve a 3

7D – El sistema muestra un mensaje indicando la causa del error.

7D1- Vuelve a 5.

9D – El sistema muestra un mensaje indicando la causa del error.

9D1- Vuelve a 7.

11D – El sistema muestra un mensaje indicando la causa del error.

11D1- Vuelve a 9.

8.3.2.1.2 Abrir Proyecto

Breve Descripción: El usuario abre un proyecto seleccionado.

Precondiciones: El proyecto debe haber sido creado anteriormente por el módulo de construcción.

Poscondiciones: Se visualizan los datos del proyecto seleccionado en el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción abrir proyecto.
- 2- El sistema pide la ruta donde se encuentra el proyecto.

- 3- El usuario llena la ruta y acepta.
- 4- El sistema abre el proyecto.

Flujos alternativos:

Cancelación.

- 3B – El usuario cancela.

8.3.2.1.3 Cerrar Proyecto

Breve Descripción: El usuario cierra el proyecto en el cual esta trabajando.

Precondiciones: -.

Poscondiciones: El mapa se encuentra si ningún proyecto cargado.

Flujo Principal:

- 1- El usuario selecciona la opción cerrar proyecto.
- 2- El sistema verifica que el proyecto se encuentra en el estado “guardado”.
- 3- El sistema cierra el proyecto.

Flujos Alternativos:

Proyecto modificado.

- 2A- El sistema verifica que el proyecto se encuentra en el estado “modificado”.
- 2A1- El sistema pregunta si quiere guardar el proyecto.
- 3A1- El usuario acepta.
- 4A1- El sistema guarda el proyecto.
- 5A1- El sistema cierra el proyecto.

Cancelación.

- 3A1B- El usuario cancela.

8.3.2.1.4 Guardar Proyecto

Breve Descripción: El usuario guarda el estado del proyecto.

Precondiciones: -.

Poscondiciones: El proyecto se encuentra en el estado guardado.

Flujo Principal:

- 1- El usuario selecciona la opción guardar proyecto.
- 2- El sistema guarda el proyecto.

8.3.2.1.5 Guardar Proyecto Como

Breve Descripción: El usuario guarda el estado del proyecto con el nombre y ruta que desee.

Precondiciones: -.

Poscondiciones: Se creó un nuevo proyecto con el nombre y ruta que ingresó el usuario.

Flujo Principal:

- 1- El usuario selecciona la opción guardar proyecto como.
- 2- El sistema pide nombre y ruta.
- 3- El usuario llena los datos y acepta.
- 4- El sistema verifica que los datos sean correctos.
- 5- El sistema crea un nuevo proyecto con los datos ingresados.

Flujos Alternativos:

Cancelación.

- 3A- El usuario cancela.

Datos incompletos.

- 3B- El usuario no llena todos los datos y acepta.
- 3B1- El sistema le indica los campos obligatorios con un mensaje.
- 5B- Vuelve a 3.

Datos incorrectos.

- 4C- El sistema emite el mensaje “Ya existe un proyecto con el mismo nombre”.
- 4C1- Vuelve a 3.

8.3.2.1.6 Salir del Sistema

Breve Descripción: El usuario desea salir de la aplicación

Precondiciones: -.

Poscondiciones: -.

Flujo Principal:

- 1- El usuario selecciona la opción salir.
- 2- El sistema finaliza su ejecución y se cierra.

Flujos Alternativos:

Guardar cambios.

2A- El sistema pide confirmación para guardar los cambios realizados.

2A1- El usuario guarda los cambios.

3A1- El sistema finaliza su ejecución y se cierra.

2A1B- El usuario descarta guardar los cambios.

3A1B1- El sistema finaliza su ejecución y se cierra.

2A1C- El usuario cancela la acción.

8.3.2.1.7 Agregar Capa de Fondo

Breve Descripción: El usuario agrega una capa de fondo al mapa.

Precondiciones: -.

Poscondiciones: Se visualizan la nueva capa en el mapa.

Flujo Principal:

- 3- El usuario selecciona la opción agregar fondo.
- 4- El sistema le pide una ruta.
- 5- El usuario selecciona la ruta.
- 6- El sistema despliega las capas que se encuentran en la ruta seleccionada.
- 7- El usuario selecciona una capa y acepta.
- 8- El sistema agrega la capa seleccionadas al mapa.

Flujos Alternativos:

Cancelación.

- 3A, 5A- El usuario cancela.

8.3.2.1.8 Quitar Capa de Fondo

Breve Descripción: El usuario quita alguna capa de fondo del mapa.

Precondiciones: -.

Poscondiciones: Se visualiza una capa de fondo menos en el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción quitar capa de fondo.
- 2- El usuario acepta
- 3- El sistema remueve la capa activa del mapa.

Flujos Alternativos:

Cancelación.

2A- El usuario cancela.

8.3.2.1.9 Full Extent

Breve Descripción: El usuario despliega el mapa en su máxima extensión.

Precondiciones: -.

Poscondiciones: Se visualiza el mapa en su totalidad.

Flujo Principal:

- 1- El usuario selecciona la opción zoom full extent.
- 2- El sistema despliega el mapa en su máxima extensión.

8.3.2.1.10 Zoom In

Breve Descripción: El usuario acerca el mapa seleccionando una posición sobre este.

Precondiciones: -.

Poscondiciones: Se visualiza el mapa a menor escala.

Flujo Principal:

- 1- El usuario selecciona la opción zoom in.
- 2- El usuario selecciona un área sobre el mapa.
- 3- El sistema acerca el mapa tomando como centro de visualización el área seleccionada.

8.3.2.1.11 Zoom Out

Breve Descripción: El usuario aleja el mapa tomando como centro de visualización una posición seleccionada por el usuario sobre este.

Precondiciones: -.

Poscondiciones: Se visualiza el mapa a mayor escala.

Flujo Principal:

- 1- El usuario selecciona la opción zoom out.
- 2- El usuario selecciona una posición del mapa.
- 3- El sistema aleja el mapa tomando como centro de visualización la posición seleccionada.

8.3.2.1.12 Pan

Breve Descripción: El usuario mueve el mapa en distintas direcciones, para visualizar las distintas partes de este.

Precondiciones: -.

Poscondiciones: Se visualiza el mapa desplazado.

Flujo Principal:

- 1- El usuario selecciona la opción pan.
- 2- El usuario selecciona una posición del mapa y mueve en una dirección manteniendo la selección.
- 3- El sistema dibuja el mapa para que se vea correctamente.

8.3.2.1.13 Información de capas

Breve Descripción: El usuario podrá visualizar la información pertinente a ciertas capas como ser las de fondo o las de entrada en formato de tabla

Precondiciones: -.

Poscondiciones: Se visualizan la tabla con la información de la capa.

Flujo Principal:

- 1- El usuario selecciona la opción información de capa.
- 2- El sistema le pide que ingrese la capa que desea visualizar, dentro de las posibles.
- 3- El usuario selecciona la capa que desee.
- 4- El usuario acepta
- 5- El sistema despliega la información de la capa en formato de tabla.

Flujos Alternativos:

Cancelación.

- 4A- El usuario cancela.

Datos incompletos.

- 5B- El sistema indica que faltan campos por completar.
- 5B1- Vuelva a 2.

8.3.2.1.14 Agregar Zona

Breve Descripción: El usuario agrega una nueva zona a la capa de zonas.

Precondiciones: -.

Poscondiciones: Se agregó una nueva zona a la capa de zonas.

Flujo Principal:

- 1- El usuario selecciona la opción agregar zona.
- 2- El usuario marca puntos sobre el mapa.
- 3- El sistema crea la zona con los puntos ingresados.
- 4- El sistema agrega la nueva zona a la capa de zonas.

8.3.2.1.15 Eliminar Zona

Breve Descripción: El usuario elimina una zona de la capa de zonas.

Precondiciones: -.

Poscondiciones: Se eliminó una zona de la capa de zonas.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar zona.
- 2- El usuario selecciona la zona en el mapa.
- 3- El sistema elimina la zona del mapa.

8.3.2.1.16 Agregar Conexión

Breve Descripción: El usuario agrega una nueva conexión a la capa de conexiones.

Precondiciones: -.

Poscondiciones: Se agregó una nueva conexión a la capa de conexiones.

Flujo Principal:

- 1- El usuario selecciona la opción agregar conexión.
- 2- El usuario marca las zonas para unir las con la nueva conexión sobre el mapa.
- 3- El sistema crea la conexión y la agrega a la capa de conexiones.

8.3.2.1.17 Eliminar Conexión

Breve Descripción: El usuario elimina una conexión de la capa de conexiones.

Precondiciones: -.

Poscondiciones: Se eliminó una conexión de la capa de conexiones.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar conexión.
- 2- El usuario selecciona la conexión en el mapa.

- 3- El sistema elimina la conexión del mapa.

8.3.2.1.18 Generar matriz origen destino

Breve Descripción: El usuario genera la matriz origen destino a partir de la información desprendida de las zonas y conexiones construidas, además de los datos sobre la demanda de una ciudad y un horizonte horario.

Precondiciones: Todos los puntos de demanda deben estar incluidos en una zona.

Poscondiciones: Se generó la matriz de demanda.

Flujo Principal:

- 1- El usuario selecciona la opción generar matriz de demanda.
- 2- El sistema pide información sobre las demandas de la ciudad. Esta información es la ruta donde se encuentra la base de datos con esta información, la tabla de viajes, el campo origen, el campo destino, el campo cantidad de viajes y el campo horas de esos viajes.
- 3- El usuario ingresa todos los campos y acepta.
- 4- El sistema le pide información acerca del horizonte horario. Esta información es la hora de comienzo del horizonte y la hora de finalización.
- 5- El usuario ingresa todos los campos y finaliza.
- 6- El sistema genera la matriz de demanda.

Cancelación.

- 3A, 5A- El usuario cancela.

Datos incompletos.

- 4B- El sistema pide se completen todos los campos.
- 4B1- Vuelve a 2.

8.3.2.1.19 Activar Notas

Breve Descripción: El sistema debe permitir activar leyendas para visualizarlas en el proyecto activo. La leyenda es información sobre las capas de entrada del proyecto o de aquellas capas que hayan sido agregadas como fondo.

Precondiciones: -.

Poscondiciones: Se activan las notas del mapa.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar activar notas del mapa.
- 2- El sistema activa las notas del mapa.

8.3.2.1.20 Desactivar Notas

Breve Descripción: El sistema debe permitir desactivar las leyendas para que no se visualicen en el proyecto activo.

Precondiciones: -.

Poscondiciones: Se desactivan las notas del mapa.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar desactivar notas del mapa.
- 2- El sistema desactiva las notas del mapa.

8.3.2.1.21 Personalizar Notas

Breve Descripción: El sistema debe permitir personalizar las leyendas. Es decir definir que campos quieren desplegar en las mismas

Precondiciones: -.

Poscondiciones: Si las notas del mapa se encuentran activas entonces se desplegarán sólo los campos personalizados.

Flujo Principal:

- 1- El usuario selecciona la opción personalizar notas.
- 2- El sistema pide los campos de las capas que se desean personalizar
- 3- El usuario selecciona la capa a personalizar y marca los campos que desea que aparezcan.
- 4- El usuario acepta

Cancelación.

- 4A- El usuario cancela.

Selecciona más capas.

- 4B- Vuelve a 3.

8.3.2.1.22 Opciones.

Breve Descripción: El sistema debe permitir distintas posibles visualizaciones sobre el mapa. Como ser el camino mínimo sobre la red de aristas, la conectividad de la red de aristas y los puntos de demanda sin zonificar.

Precondiciones: Existe un proyecto abierto.

Poscondiciones: Se visualiza el camino mínimo sobre la red de aristas y/o la conectividad de la red de aristas y/o los puntos de demanda sin zonificar sobre el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción Opciones.
- 2- El sistema pide se marquen las opciones de camino mínimo sobre la red de aristas y/o la conectividad de la red de aristas y/o los puntos de demanda sin zonificar
- 3- El usuario selecciona las opciones de camino mínimo sobre la red de aristas y/o la conectividad de la red de aristas y/o los puntos de demanda sin zonificar
- 4- El usuario acepta

Cancelación.

- 3A, 4A- El usuario cancela.

8.3.2.2 Casos de Uso - Módulo de Manipulación del Caso de Estudio

8.3.2.2.1 Nuevo Proyecto

Breve Descripción: El sistema debe permitir crear un nuevo proyecto de trabajo.

Precondiciones: -.

Poscondiciones: Se creó un nuevo proyecto y está cargado en el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción nuevo proyecto.
- 2- El sistema pide el nombre del proyecto, la ruta donde se desea guardar, y da la opción de generar un nuevo proyecto a partir del módulo de construcción.
- 3- El usuario completa los datos y no marca la opción de generar un nuevo proyecto a partir del módulo de construcción.
- 4- El usuario selecciona siguiente.
- 5- El sistema pide las rutas de una capa de aristas, de una capa de puntos de demanda y de una matriz de demanda.

6- El usuario finaliza.

7- El sistema crea el proyecto y lo carga en el mapa.

Cancelación.

4A, 6A- El usuario cancela.

Datos incompletos.

3C- El usuario no completa todos los datos.

3C1- El usuario selecciona siguiente.

3C2- El sistema pide que se completen los datos.

3C3- El usuario acepta.

3C4- Vuelva a 2

5C- El usuario no completa todos los datos.

5C1- El usuario selecciona siguiente.

5C2- El sistema pide que se completen los datos.

5C3- El usuario acepta.

5C4- Vuelva a 4.

Crear proyecto a partir del módulo de construcción.

3B- El usuario completa los datos y marca la opción de generar un nuevo proyecto a partir del módulo de construcción.

3B1- El sistema pide la ruta del proyecto generado con el módulo de construcción.

3B2 – El usuario finaliza.

3B3- El sistema crea el proyecto y lo carga en el mapa.

Cancelación.

3B2A- El usuario cancela.

Datos incompletos.

3BC- El usuario no completa todos los datos.

3BC1- El usuario selecciona siguiente.

3BC2- El sistema pide que se completen los datos.

3BC3- El usuario acepta.

3BC4- Vuelva a 2

3B1C- El usuario no completa todos los datos.

3B1C1- El usuario selecciona siguiente.

3B1C2- El sistema pide que se completen los datos.

3B1C3- El usuario acepta.

3B1C4- Vuelva a 3B1.

8.3.2.2.2 Abrir Proyecto

Igual que Abrir Proyecto módulo construcción.

8.3.2.2.3 Cerrar Proyecto

Igual que Cerrar Proyecto módulo construcción.

8.3.2.2.4 Guardar Proyecto

Igual que Guardar Proyecto módulo construcción.

8.3.2.2.5 Guardar Proyecto Como

Igual que Guardar Proyecto Como módulo construcción.

8.3.2.2.6 Cargar Fondo

Igual que cargar fondo del módulo construcción.

8.3.2.2.7 Quitar Fondo

Igual que quitar fondo del módulo construcción.

8.3.2.2.8 Editar Recorrido Activo

Breve Descripción: El usuario realiza cambios al recorrido activo.

Precondiciones: -.

Poscondiciones: Se visualizan los cambios en el recorrido.

Flujo Principal:

- 1- El usuario selecciona la opción editar recorrido.
- 2- El usuario modifica el recorrido. Modificar puede ser agregar o quitar tramos, o cambiar la frecuencia del recorrido.

- 3- El sistema despliega las modificaciones.
- 4- El usuario selecciona fin editar recorrido.
- 5- El sistema finaliza la edición.

Flujos alternativos:

Ningún cambio.

3A- El usuario selecciona fin editar recorrido.

Realizar más modificaciones.

4B- El usuario vuelve a 2.

8.3.2.2.9 Agregar Tramo

Breve Descripción: El usuario agrega un tramo al recorrido activo.

Precondiciones: El usuario se encuentra trabajando sobre una solución que contiene al menos un recorrido.

Poscondiciones: Se visualiza el tramo agregado en el recorrido activo.

Flujo Principal:

- 1- El usuario selecciona la opción agregar tramo.
- 2- El usuario selecciona la conexión por la que desea que el recorrido tenga un tramo.
- 3- El sistema despliega el nuevo tramo.

8.3.2.2.10 Eliminar Tramo

Breve Descripción: El usuario elimina un tramo del recorrido activo.

Precondiciones: El usuario se encuentra trabajando sobre una solución que contiene al menos un recorrido.

Poscondiciones: Se visualiza el tramo eliminado del recorrido activo. (No se ve más ese tramo)

Flujo Principal:8

- 1- El usuario selecciona la opción eliminar tramo.
- 2- El usuario selecciona el tramo que desea eliminar del recorrido activo.
- 3- El sistema muestra el mapa sin el tramo eliminado.

8.3.2.2.11 Modificar Frecuencia.

Breve Descripción: El usuario modifica la frecuencia del recorrido activo.

Precondiciones: El usuario se encuentra trabajando sobre una solución que contiene al menos un recorrido.

Poscondiciones: Se visualiza el cambio de frecuencia en el recorrido activo.

Flujo Principal:

- 1- El usuario selecciona la opción modificar frecuencia.
- 2- El sistema pide que se ingrese la frecuencia deseada.
- 3- El usuario ingresa la frecuencia.
- 4- El usuario acepta
- 5- El sistema refleja el cambio en el mapa.

Flujos alternativos:

Cancelación.

- 4A- El usuario cancela.

8.3.2.2.12 Full Extent

Igual que Full Extent del módulo construcción.

8.3.2.2.13 Zoom In

Igual que Zoom In del módulo construcción.

8.3.2.2.14 Zoom Out

Igual que Zoom Out del módulo construcción.

8.3.2.2.15 Pan

Igual que Pan del módulo construcción.

8.3.2.2.16 Abrir Solución

Breve Descripción: El sistema le permite al usuario abrir una solución previamente creada.

Precondiciones: Existe al menos una solución en el proyecto que se encuentra cargado en el mapa.

Poscondiciones: Se abrió la solución y se visualiza en el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción abrir solución.
- 2- El sistema pide se seleccione la solución a ser abierta dentro de las existentes.
- 3- El usuario selecciona una solución.
- 4- El usuario acepta.
- 5- El sistema carga la solución en el mapa.

Flujos alternativos:

Cancelación.

4A- El usuario cancela

Datos incompletos.

5B- El sistema solicita se completen los datos.

5B1- El usuario acepta

5B2- Vuelve a 2

8.3.2.2.17 Cerrar Solución

Breve Descripción: El sistema le permite al usuario cerrar una solución.

Precondiciones: Existe una solución en cargada en el mapa.

Poscondiciones: Se cerró la solución y no se visualiza en el mapa.

Flujo Principal:

- 1- El usuario selecciona la opción cerrar solución.
- 2- El sistema pide se confirme guardar los cambios
- 3- El usuario afirma guardar los cambios.
- 4- El sistema descarga la solución del mapa y la guarda.

Flujos alternativos:

Cancelación.

3A- El usuario cancela

No se guardan los cambios.

3B- El usuario niega guardar los cambios

3B1- El sistema descarga la solución del mapa y no la guarda.

No hubo cambios.

2C- El sistema descarga la solución del mapa.

8.3.2.2.18 Generar Solución

Breve Descripción: El usuario ejecuta los algoritmos que generan los recorridos solución.

Precondiciones: -.

Poscondiciones: Se generan las soluciones propuestas por el algoritmo.

Flujo Principal:

- 1- El usuario selecciona la opción generar solución.
- 2- El sistema pide nombre de la solución y la configuración de frecuencia y capacidad a utilizar.
- 3- El usuario completa los datos.
- 4- El usuario selecciona siguiente
- 5- El sistema pide se elija el algoritmo a ejecutar de entre los que están disponibles.
- 6- El usuario selecciona un algoritmo.
- 7- El usuario selecciona siguiente.
- 8- El sistema pide se completen los valores de los parámetros requeridos.
- 9- El usuario completa los valores de los parámetros.
- 10- El usuario finaliza.
- 11- El sistema genera las soluciones resultantes de ejecutar los algoritmos.

Flujos alternativos:

Cancelación.

4A, 7A, 10A - El usuario cancela

Datos incompletos.

5B- El sistema solicita se completen los datos.

5B1- El usuario acepta

5B2- Vuelve a 2

8B- El sistema solicita se completen los datos.

8B1- El usuario acepta

8B2- Vuelve a 5

11B- El sistema solicita se completen los datos.

11B1- El usuario acepta

11B2- Vuelve a 8

8.3.2.2.19 Evaluar Recorridos

Breve Descripción: El usuario evalúa los recorridos sobre la red utilizando los algoritmos.

Precondiciones: Existe un proyecto con una solución cargada en el mapa.

Poscondiciones: Se generan los datos de la evaluación.

Flujo Principal:

5- El usuario selecciona la opción evaluar recorridos.

6- El sistema pide se seleccione el algoritmo a utilizar.

7- El usuario selecciona un algoritmo.

8- El usuario selecciona siguiente.

9- El sistema pide los valores de los parámetros que se necesitan completar

10- El usuario completa los valores de los parámetros.

11- El usuario selecciona finalizar.

12- El sistema genera la evaluación de la solución.

Flujos alternativos:

Cancelación.

4A, 6A- El usuario cancela.

Datos incompletos

5B- El sistema pide se completen los campos.

5B1- El usuario acepta.

5B2- Vuelve a 2

8B- El sistema pide se completen los campos.

8B1- El usuario acepta.

8B2- Vuelve a 5

8.3.2.2.20 Graficar evaluación

Breve Descripción: El usuario grafica la evaluación de la solución.

Precondiciones: Se corrió exitosamente la evaluación de una solución.

Poscondiciones: Se vio el resultado de la evaluación de una solución en formato de gráfica.

Flujo Principal:

- 1- El usuario selecciona la opción gráfica.
- 2- El sistema despliega la última evaluación realizada en formato de gráfica.

8.3.2.2.21 Activar Notas

Igual que Activar Notas del módulo construcción.

8.3.2.2.22 Desactivar Notas

Igual que Desactivar Notas del módulo construcción.

8.3.2.2.23 Personalizar Notas

Igual que Personalizar Notas del módulo construcción.

8.3.2.2.24 Información de capas

Breve Descripción: El usuario podrá visualizar la información pertinente a ciertas capas como ser las de fondo en formato de tabla

Precondiciones: -.

Poscondiciones: Se visualizan la tabla con la información de la capa.

Flujo Principal:

- 1- El usuario selecciona la opción información de capa.
- 2- El sistema le pide que ingrese la capa que desea visualizar, dentro de las posibles.
- 3- El usuario selecciona la capa que desee.
- 4- El usuario acepta
- 5- El sistema despliega la información de la capa en formato de tabla.

Flujos Alternativos:

Cancelación.

4A- El usuario cancela.

Datos incompletos.

5B- El sistema indica que faltan campos por completar.

5B1- Vuelva a 2.

8.3.2.2.25 Crear configuración de frecuencias y capacidad.

Breve Descripción: El sistema debe permitir crear un conjunto de frecuencias y capacidades aplicables a cualquier solución.

Precondiciones: -.

Poscondiciones: Se crea una configuración con un conjunto de frecuencias y capacidades.

Flujo Principal:

- 1- El usuario selecciona la opción crear configuración de frecuencias y capacidades.
- 2- El sistema le pide que ingrese nombre de la configuración, capacidad máxima por ómnibus y cardinalidad del conjunto discreto de frecuencias.
- 3- El usuario completa los campos y selecciona siguiente.
- 4- El sistema pide que ingrese los valores para los elementos del conjunto de frecuencias.
- 5- El usuario acepta.

Flujos Alternativos:

Cancelación.

3A, 5A- El usuario cancela.

Datos incompletos.

4B- El sistema indica que faltan campos por completar.

4B1- Vuelva a 2.

8.3.2.2.26 Eliminar configuración de frecuencias y capacidad.

Breve Descripción: El sistema debe permitir eliminar un conjunto de frecuencias y capacidades aplicables a cualquier solución.

Precondiciones: Existe al menos una configuración de frecuencias y capacidades.

Poscondiciones: Se elimina una configuración de frecuencias y capacidades.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar configuración de frecuencias y capacidades.
- 2- El sistema le pide que seleccione el nombre de la configuración
- 3- El usuario selecciona.
- 4- El usuario acepta.

Flujos Alternativos:

Cancelación.

- 3A, 4A- El usuario cancela.

Datos incompletos.

- 4B- El sistema indica que faltan campos por completar.
- 4B1- Vuelva a 2.

8.3.2.2.27 Vista recorridos.

Breve Descripción: El sistema debe permitir ver recorridos en el mapa diferenciados de acuerdo a la frecuencia y a la dirección como atributos visuales.

Precondiciones: Existe un proyecto abierto con una solución cargada.

Poscondiciones: Se visualizan los recorridos de acuerdo a la opción elegida.

Flujo Principal:

- 1- El usuario selecciona la opción eliminar vista recorridos
- 2- El sistema le pide que seleccione la vista normal o la ponderada por frecuencia
- 3- El usuario selecciona la vista normal.
- 4- El usuario acepta.

Flujos Alternativos:

Cancelación.

- 3A, 4A- El usuario cancela.

Vista ponderada por frecuencia.

- 3B- El sistema selecciona la vista ponderada por frecuencia.
- 3B1- Vuelve a 4.

8.4 *Arquitectura de la Herramienta*

La arquitectura a grandes rasgos fue descrita en el informe en la sección 2.3 (Diseño y arquitectura). En este anexo se mostrará un diagrama de estructura estática de los módulos de construcción y manipulación del caso de estudio. El nivel de granularidad de estos no llega a las funciones mismas, pero servirá para transmitir la idea llevada a cabo a la hora de implementar la solución.

Como habíamos visto en la sección 2.3 ambos módulos están estructurados según dos componentes:

- componente geográfico: maneja la información geográfica del módulo
- componente central: maneja toda la lógica del problema, interactúa con el usuario, con el componente geográfico, se encarga del acceso a datos, la persistencia y la comunicación con los algoritmos en el caso del módulo de manipulación.

Debido a las características de ambos componentes, el geográfico tendrá un diseño tipo biblioteca, brindando sus funcionalidades al componente central el cual se encargará de toda la lógica del negocio. Este componente central tiene una arquitectura de capas.

Extraído de [AC 2008] .

Programación por capas

Definición: Es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos.

La ventaja principal: El desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, la repercusión del mismo estará acotada.

- Además, permite distribuir el trabajo de creación de una aplicación por niveles; cada grupo de trabajo está totalmente abstraído del resto de los niveles, de forma que basta con conocer la *API* que existe entre niveles.

API (Application Programming Interface):

Es el conjunto de funciones y procedimientos o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Capa de presentación.

Esta capa es la que ve el usuario, le presenta el sistema, le comunica la información y captura la información del usuario en un mínimo de proceso.

Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" para el usuario, generalmente se presentan como formularios.

Capa de negocio (o capa lógica)

Aquí es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

- Toda aplicación tiene código para implementar reglas de negocios.
- Se puede seleccionar almacenar la lógica de negocios sobre cada estación de cliente, u optar por ejecutar la lógica de negocios sobre un servidor de aplicaciones.
- No toda la lógica de negocio es la misma, algunas no requieren un frecuente acceso a los datos, pero una interfaz de usuario robusta necesitará de la lógica de negocios para la validación en la entrada de campos, cálculos en tiempo real u otras interacciones de usuarios.

Capa de datos

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Diferencia entre Capas y Niveles

Capa.- Forma como una solución es segmentada desde el punto de vista lógico.

Nivel.- Forma en que las capas lógicas se encuentran distribuidas de forma física.

Ejemplos

1. Tres capas que residen en un ordenador (Presentación + lógica + datos)
2. Tres capas que residen en dos ordenadores (Presentación + lógica, datos).
3. Tres capas que residen en tres ordenadores (Presentación, lógica, datos).

Quizás la desventaja más grande del modelo de capas es que empeora la performance de la aplicación en general, ya que existe un tiempo perdido en la coordinación entre las

distintas capas. Pero como al no ser el tiempo de respuesta uno de los requisitos prioritarios en este proyecto, optamos por aplicar una arquitectura en capas.

El siguiente diagrama (figura 8.4.1) aplica tanto para el módulo de construcción del caso de estudio como para el de manipulación.

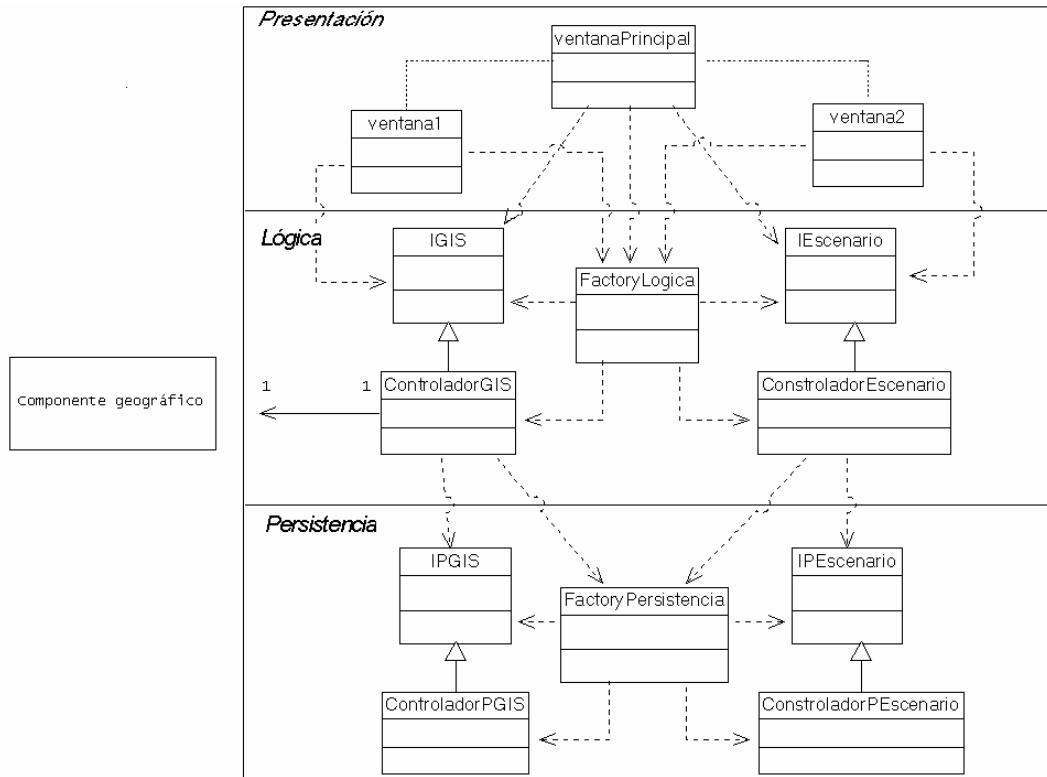


Figura 8.4.1.

8.4.1 Funcionalidades de las clases.

Presentación.

- ventanaPrincipal: Es el punto de acceso para el usuario, hacia todas las funcionalidades del sistema. Es la ventana principal.
- ventanaN: Se encarga de facilitarle al usuario el acceso a alguna funcionalidad específica. Es invocada por parte de la ventana principal

Lógica.

- FactoryLogica: Es el punto de acceso a las interfaces que brinda la capa lógica.

- IGIS: Es la interfaz que brinda las funcionalidades de manipulación de información geográfica a la capa de presentación. Las funcionalidades son todas aquellas que involucran al mapa.
- IEscenario: Es la interfaz que proporciona las funcionalidades de control del proyecto sobre el cual se está trabajando. Creación y edición del mismo.
- ControladorEscenario: Se encarga de implementar las funcionalidades especificadas en IEscenario.
- ControladorGIS: Se encarga de implementar las funcionalidades especificadas en IGIS.

Persistencia.

- FactoryPersistencia: Es el punto de acceso a las interfaces que brinda la capa de persistencia.
- IPGIS: Es la interfaz que brinda las funcionalidades de persistencia de la información geográfica a la capa lógica. (Persistencia de capas)
- IPEscenario: Es la interfaz que brinda las funcionalidades de persistencia del proyecto activo a la capa lógica. (Persistencia de rutas del proyecto)
- PGIS: Se encarga de implementar las funcionalidades especificada en IPGIS.
- PEscenario: Se encarga de implementar las funcionalidades especificadas en IPEscenario.

9. Anexo. Verificación y Validación

Durante el proceso de desarrollo hemos realizado pruebas de caja blanca y revisión por pares. Posteriormente los ensayos fueron de caja negra los cuales serán descritos en este anexo.

Todas las pruebas fueron realizadas desde los distintos puntos de acceso a las funcionalidades que presenta la herramienta, es decir desde los menús o desde la barra de herramientas.

9.1 Módulo Central

- Escenario: Inicio de todos los módulos.

	Pasos seguidos	Correcto/Falla
1.	Levantar la aplicación igoR-tp.	C
2.	Levantar el Módulo de Manipulación.	C
3.	Levantar el Módulo de Construcción.	C
4.	Levantar el Módulo de Algoritmos.	C
5.	Cerrar los tres módulos sucesivamente.	C
6.	Invocar los contenidos de la ayuda.	C
7.	Acceder al menú “Sobre el Programa”	C
8.	Cerrar igoR-tp	C

9.2 Módulo Configuración de Algoritmos

- Escenario: Creación de archivo configuración de algoritmos.

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Algoritmos.	C
2.	Seleccionar la opción “Crear archivo de configuración del algoritmo”.	C
3.	Siguiente.	C
4.	Ubicación:...\igor-tpv1.0\OptimEx\OptimEx.exe	C

	Cantidad de parámetros: 3	
5.	Siguiente	C
6.	Se completan los parámetros de esta forma: (param1, caracteres); (param2, entero);(param3,real)	C
7.	Finalizar	C

- Escenario: Eliminar archivo de configuración de Algoritmos

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Algoritmos.	C
2.	Seleccionar la opción “Eliminar archivo de configuración del algoritmo”.	C
3.	Siguiente.	C
4.	Seleccionar el archivo conf_OptimEx.exe.xml	C
5.	Finalizar	C

9.3 Módulo Construcción del Caso de Estudio

- Escenario: Crear nuevo proyecto.

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de construcción.	C
2.	Seleccionar el menú Proyecto/Nuevo	C
3.	Nombre del proyecto: prueba1 Ubicación: C:\Ejemplo	C
4.	Siguiente	C
5.	Capa puntos de demanda: ...\\igoR-tpv1.0\Datos\Centroides_MANZANAS.shp Id de los puntos de demanda: ID	C
6.	Siguiente	C
7.	Capa de aristas: ...\\igoR-tpv1.0\Datos\EJES.shp Marcar la opción “calcular red de aristas”	C

8.	Siguiente	C
9.	Factor Conversor: 1 (la capa ya está en metros) Velocidad media: 45	C
10.	Finalizar	C

- Escenario alternativo: Crear nuevo proyecto sin calcular la red de aristas

7b.	Capa de aristas: ..\igoR-tpv1.0\Datos\EJES.shp Desmarcar la opción “calcular red de aristas” Campo ‘From: fromNodo Campo ‘To’: tonado	C
-----	--	---

- Escenario alternativo: Crear nuevo proyecto con capa de aristas que ya contenga factor conversor conocido para la aplicación.

9b.	Velocidad media: 45	C
-----	---------------------	---

- Escenario: Apertura proyecto, creación de zonas, conexiones y generación de la matriz demanda

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de construcción.	C
2.	Seleccionar el menú Proyecto/Abrir	C
3.	Ubicación:...\igoR-tpv1.0\Datos\prueba1\prueba1.construccion Desmarcar la opción recalcular la red de aristas	C
4.	Aceptar	C
5.	Seleccionar el menú Zona/ Agregar Zona	C
6.	Marcar una zona en el mapa que incluya al menos un punto de demanda	C
7.	Repetir el paso 6 hasta que cada uno de los puntos de demanda quede incluido en una zona.	C
8.	Seleccionar el menú Conexión/ Agregar Conexión	C
9.	Seleccionar las zonas de a pares para conectarlas	C

10.	Repetir el paso 9 tantas veces se crea necesario.	C
11.	Seleccionar el menú Matriz demanda/ Generar	C
12.	Completar los campos de esta forma: Base de datos: ...\\RiveraTransporte.mdb Tabla de viajes: ViajesString Campo origen: origen Campo destino: destino Cantidad de viajes: nroViaje Campo hora: hora	C
13.	Siguiente	C
14.	Se completan las horas de inicio y fin. Inicio: 00:00 Fin: 12:15	C
15.	Finalizar	

9.4 *Módulo Manipulación del Caso de Estudio*

- Escenario: Crear nuevo proyecto.

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Manipulación.	C
2.	Seleccionar el menú Proyecto/ Nuevo	C
3.	Completar los campos. Nombre: prueba2 Ubicación: C:\\Ejemplo No marcar la opción “Crear nuevo proyecto a partir del módulo construcción”	C
4.	Siguiente	C
5.	Completar los campos solicitados. Capa de centroides: ...\\igoR-tpv1.0\\Datos\\prueba1\\prueba1_Centroides.shp Capa de conexiones: ...\\igoR-	C

	tpv1.0\Datos\prueba1\prueba1_Conexiones.shp Matriz demanda: ... \igoR- tpv1.0\Datos\prueba1\MatrizDemanda.xml	
6.	Finalizar	C

- Escenario alternativo: Crear nuevo proyecto a partir del módulo de construcción.

3b.	Completar los campos. Nombre: prueba2 Ubicación: C:\Ejemplo Marcar la opción “Crear nuevo proyecto a partir del módulo construcción”	C
3b1.	Siguiente	C
3b2.	Completar la ubicación del proyecto generado con el módulo de construcción. Ubicación: ... \igoR- tpv1.0\Datos\prueba1\prueba1.construccion	C
3b3.	Finalizar	C

- Escenario: Crear recorrido y nueva solución.

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Manipulación.	C
2.	Seleccionar el menú Proyecto/ Abrir	C
3.	Ubicación: ... \igoR- tpv1.0\Datos\prueba2\prueba2.manipulacion	C
4.	Aceptar	C
5.	Seleccionar el menú Recorridos/ Nuevo	C
6.	Seleccionar la opción <Nueva Solución...>	C
7.	Aceptar	C

8.	Completar los campos. Nombre solución: solucionPrueba Cantidad de pasajeros por bus: 40 Cantidad de frecuencias: 3	C
9.	Siguiente	C
10.	Completar las frecuencias por hora: Frecuencia1: 5 Frecuencia2: 10 Frecuencia3: 15	C
11.	Siguiente	C
12.	Completar el nombre del recorrido: recorrido1	C
13.	Finalizar	C

- Escenario alternativo: Crear recorrido en una solución existente.

6b.	Seleccionar la una solución ya creada. Solución: solucionPrueba.	C
6b1.	Aceptar	C
6b2.	Completar el nombre del recorrido: recorrido2	C
6b3.	Finalizar	C

- Escenario alternativo: Crear recorrido en una solución abierta.

5b.	Seleccionar el menú Solución/ Abrir Solución	C
6.	Seleccionar la una solución ya creada. Solución: solucionPrueba.	C
7.	Aceptar	C
8.	Seleccionar el menú Recorridos/ Nuevo	C
9.	Completar el nombre del nuevo recorrido: recorrido3	C
10.	Aceptar	C

- Escenario: Generar solución con algoritmo de optimización y desplegarla

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Manipulación.	C
2.	Seleccionar el menú Proyecto/ Abrir	C
3.	Ubicación: ... \igoR-tpv1.0\Datos\prueba2\prueba2.manipulacion	C
4.	Aceptar	C
5.	Seleccionar el menú Solución/ Generar Recorridos	C
6.	Completar los campos. Nombre solución: solucionAlgoritmo Configuración de frecuencias y capacidad: Conf1	
9.	Siguiente	C
10.	Seleccione el algoritmos a emplear. Ejemplo: conf_Optimex.exe.xml	C
11.	Aceptar	C
12.	Completar los parámetros del algoritmo Param1: a Param2: 2 Param3: 3.0	C
13.	Aceptar	C

- Escenario Evaluar solución con algoritmos de evaluación y ver gráfica.

	Pasos seguidos	Correcto/Falla
1.	Levantar el módulo de Manipulación.	C
2.	Seleccionar el menú Proyecto/ Abrir	C
3.	Ubicación: ... \igoR-tpv1.0\Datos\prueba2\prueba2.manipulacion	C
4.	Aceptar	C
5.	Seleccionar el menú Solución/ Evaluar Recorridos	C
6.	Seleccione el algoritmo a emplear. Ejemplo: conf_EvalEx.exe.xml	C

7.	Aceptar	C
8.	Completar los parámetros del algoritmo evalParam: eval	C
9.	Aceptar	C
10.	Seleccionar el recorrido que se desea ver las gráficas	C
11.	Seleccionar el menú Herramientas/Gráfica	C

Si bien realizamos distintos tipos de pruebas que mostraron que el sistema se encuentra estable la retroalimentación más importante fue la del propio usuario.

Además mediante la misma se pudo evaluar no sólo las funcionalidades sino también la facilidad de uso de la herramienta. A continuación agregamos algunos de sus comentarios, los cuales fueron abordados como se verá entre líneas:

Errores detectados.

Módulo Configuración algoritmos:

- Error al salvar configuración (no encuentra un archivo del notificador).
 - Solucionado

Módulo de Construcción:

- La primera vez que inicio el Módulo de construcción da un error.
 - Solucionado.
- Centroides y conexiones no son parte del fondo.
 - Solucionado.

Módulo Manipulación:

- Herramientas, información del mapa: la lista de información de la capa está vacía. (Deshabilito los controles si no hay datos para mostrar pero también se deshabilitan las notas del mapa)
 - Solucionado

- Si creo un proyecto de manipulación a partir de uno de construcción no lo salva (aparentemente, verificarlo).
 - Solucionado

General:

- Cuando cierro el cuadro de notificación da una excepción.
 - Solucionado.

Solicitudes de cambios.

Módulo Configuración algoritmos:

- Para cada parámetro permitir ingresar una observación que al momento de invocar el algoritmo se muestre
 - Agregado
- Sería bueno poder crear configuraciones de frecuencias y capacidades de buses para no tener que ingresarlas cada vez.
 - Agregado

Módulo Construcción:

- Información del mapa, demanda: poner todos los valores con la misma cantidad de cifras decimales; indicar unidades.
 - Agregado
- Ver el total de demanda en alguna parte.
 - Agregado.
- Información de demanda: dibujar más grande los cuadros sobre los vértices del grafo.
 - Modificado.
- Historial de mensajes del asistente en una corrida.
 - Se encontró una alternativa válida para que el usuario pueda ver la información que necesita mediante la tabla que despliega la matriz de demanda
- Si no tengo la matriz creada no puede habilitar la opción información de capas (demanda).

- Modificado
- ¿Que hace: ver demanda no zonificada?
 - Modificado. Se describió mejor lo que indica la etiqueta. Ver los puntos de demanda que no están incluidos en ninguna zona.

•

Módulo Manipulación:

- En cuadro de ingreso de frecuencias, aclarar que son tiempos entre arribos (y las unidades).
 - Modificado
- Directos y transbordos: la tabla de transbordos no se visualiza fácilmente.(La situación era la visibilidad de la tabla de transbordos).
 - Modificado. Se puso las dos tablas en una ventana con un separador al estilo ventana principal
- Habría que dibujar solo los trasbordos que seleccione. Esto requiere un poco de trabajo.
 - Agregado.
- Herramientas/Gráfica: debería llamarse "Histograma de Perfil de carga".
 - Modificado.
- ¿Cómo funciona Eliminar Tramo?
 - Modificado. Se agregó mensaje explicativo cuando realiza CLICK pero no se selecciona nada.
- Al seleccionar un algoritmo de generación o evaluación, presenta los nombres de los ejecutables (sería mejor tener identificadores).
 - No se ha modificado, queda para trabajos a futuro.

En general:

- Siempre que se pida un dato numérico cuyas unidades son relevantes, indicar cuales son esas unidades al usuario.
 - Agregado
- Verificar caminos absolutos y relativos de archivos. Avisar consecuencias en el manual de usuario (por ejemplo si se copia un proyecto para abrirlo en otra instalación).
 - Agregado.

- Cambiar tamaño de la barra izquierda.
 - Agregado.
- Información de la capa: no se distingue si está con un zoom alejado.
Sugerencia: hacer zoom al lugar del objeto seleccionado en la tabla.
 - Modificado.
- Notas del mapa no desaparecen.
 - Modificado. Se agregó a nivel de código un reloj que permite que las notas del mapa se refresquen correctamente cuando el cursor del ratón se detiene sobre un punto específico, evitando el efecto producido cuando el puntero se mueve continuamente
- Herramientas: agrupar opciones de Notas del mapa.
 - Modificado.
- ¿Qué es Bar escalas?

Era un menú obsoleto que fue quitado. El desarrollo de esta funcionalidad puede ser un trabajo a futuro.

Otras consideraciones que se tuvo en cuenta una vez que el usuario probó la aplicación, fueron el tamaño tanto de la fuente de los menús como de los íconos de la barra de herramientas. En versiones anteriores no habíamos notado el efecto que producía si iGoR-tp corría en monitores de baja resolución. Por lo tanto adecuamos esos tamaños a las necesidades del usuario.

10. Anexo. Prototipos

En este anexo se verá el proceso de evolución de los prototipos realizados primero en papel y luego en máquina. Se detalla el trabajo hecho para el módulo de construcción del caso de estudio, siendo el proceso muy similar para el de manipulación del caso de estudio.

10.1 Menús

Proyecto	<i>Vista</i>	<i>Imagen</i>	<i>Zonas</i>	<i>Solución</i>	<i>Ayuda</i>
	<i>Nuevo</i> <i>Abrir</i> <i>Cerrar</i> <i>Guardar</i> <i>Guardar Como</i> <i>Salir</i>				

<i>Proyecto</i>	Vista	<i>Imagen</i>	<i>Zonas</i>	<i>Solución</i>	<i>Ayuda</i>
	<i>Zoom In</i> <i>Zoom Out</i> <i>Full Extent</i> <i>Pan</i> <i>Bar Escalas</i>				

<i>Proyecto</i>	Vista	<i>Imagen</i>	<i>Zonas</i>	<i>Solución</i>	<i>Ayuda</i>
		<i>Cargar Imagen</i> <i>Remover Imagen</i>			

<i>Proyecto</i>	<i>Vista</i>	<i>Imagen</i>	Zonas	<i>Solución</i>	<i>Ayuda</i>
				<i>Nueva Capa</i> <i>Guardar Capa</i> <i>Agregar Zona</i> <i>Eliminar Zona</i> <i>Expandir Zona</i> <i>Contraer Zona</i>	

<i>Proyecto</i>	<i>Vista</i>	<i>Imagen</i>	<i>Zonas</i>	Solución	<i>Ayuda</i>
				<i>Generar Grafo</i> <i>Agregar Conexiones</i> <i>Eliminar Conexiones</i>	

<i>Proyecto</i>	<i>Vista</i>	<i>Imagen</i>	<i>Zonas</i>	<i>Solución</i>	Ayuda
					<i>Ayuda Prog</i> <i>Sobre Prog</i>

10.2 Escenario de Interacción

Este escenario fue descrito en la etapa del prototipo en papel, por esta razón es posible que el mismo varíe con la última versión de igoR-tp.

El cliente desea zonificar, utilizando un criterio propio, a la ciudad de Rivera. Además, a partir de la zonificación desea crear un grafo que tenga como vértices a los centroides de las zonas y como aristas conexiones entre estos centroides. Cuenta con

una capa de poli-líneas para las calles llamada “Calles_Rivera.shp”, esta capa de datos representan la red vial de la ciudad, en definitiva es la capa de aristas. También cuenta con una capa puntos para los puntos demanda llamada “Puntos_Demanda_Rivera.shp”. Cada zona que Antonio (el usuario) cree contendrá puntos demanda, estos puntos demanda se utilizarán para calcular la demanda de la zona.

Para poder llevar a cabo su propósito Antonio inicia el sistema creado por el proyecto de grado 2008 de Diego Gawenda y Manuel Martínez. Luego de ejecutar el programa Antonio visualiza ciertas opciones en el menú. Dentro de estas se encuentra Proyecto, ingresa y visualiza entonces Nuevo. Antonio selecciona esta opción y entonces el sistema le solicita que ingrese el nombre para el proyecto, una ruta del sistema de archivos, una capa de poli-líneas para las aristas de la red y una capa de puntos para los puntos demanda. Antonio ingresa todos los datos. El proyecto “Proyecto_Rivera” se crea (Figura 1).

Antonio quiere ahora comenzar con la construcción de las zonas. Para esto visualiza en el menú la opción Zonas. El usuario se dispone a comenzar a editar la capa creada de zonas, para esto ingresa en el menú Zonas y selecciona Nueva Zona. Ahora marca 4 puntos sobre el mapa formando un polígono de manera de encerrar todos los puntos demanda que desea. Antonio visualiza entonces que en el mapa aparece una zona, llamémosla A, que encierra todas los puntos demanda deseados (Figura 2). Entonces Antonio se propone a crear otra zona. Para eso realiza las mismas selecciones que para la zona que creó y vuelve a marcar los 4 puntos, creando así una nueva zona, la zona B. Nuevamente Antonio decide ingresar una nueva zona. Al ingresar las marcas sobre el mapa, Antonio se percata de que el polígono ingresado con las marcas encierra puntos demanda del área A, pero visualiza en el mapa que la nueva área C es contigua a la zona A pero no se intersectan (Figura 3). Mientras Antonio continúa con la edición de la capa aparece Marita (otro usuario). Marita observa que la zona B esta encerrando más puntos demanda de los acordados. Entonces le pide a Antonio que haga algo al respecto. Antonio va a la opción Zonas y ve que esta la opción Eliminar Zona, la selecciona. Luego selecciona la zona B y ya no se visualiza más (Figura 4). Entonces Marita le pide a Antonio que ingrese dos nuevas zonas D y E (Figura 5). Antonio las ingresa. Marita entonces decide ingresar la última zona del caso, la F, para esto encierra con un polígono los puntos de demanda que no están encerrados. En el proceso también

encierra puntos demanda que pertenecen a las zonas A, C, D y E, pero una vez finalizado el ingreso de las marcas del polígono se visualiza que la nueva zona F no contiene ningún punto demanda de otra zona y es contigua a A, D y E (Figura 6). Ahora ambos deciden que es momento de generar el grafo solución del caso de estudio de la Ciudad de Rivera. Para esto van a la opción Grafo y seleccionan Generar Grafo, entonces se visualiza que aparecen nuevos puntos y líneas en el mapa. Estos puntos son los centroides de las zonas y las líneas son las conexiones entre los centroides. Se puede visualizar que solo existen conexiones entre centroides pertenecientes a zonas adyacentes (Figura 7). Mientras que Marita está contenta con el resultado, Antonio se percata que en realidad hay una conexión que no es útil, la conexión (f, e). Luego de discutirlo con Marita deciden eliminar esta conexión. Para esto seleccionan la opción Grafo del menú y luego seleccionan Eliminar Conexión. Luego seleccionan la conexión (f, e) y esta desaparece del mapa. Ahora si, ambos se sienten satisfechos con el arduo trabajo realizado y deciden guardar lo hecho hasta el momento. Seleccionan Proyecto del menú y luego seleccionan Guardar. Es así como finaliza este escenario de interacción.

Luego del arduo trabajo realizado con el modulo de preparación del caso de estudio. Antonio y Marita deciden probar sus algoritmos a modo de comparación. Desde hace tiempo cada uno de ellos se ha vanagloriado que su algoritmo es el mejor y deciden que es tiempo de probarlo, como se dice popularmente "...los pingos se ven en la pista...". Para esto inician el módulo de manipulación del caso de estudio del proyecto de grado 2008 desarrollado por Diego Gawenda y Manuel Martínez. Al ver las opciones que esta herramienta ofrece, notan que es similar a la anterior ya que tiene unas cuantas opciones que se repiten como Proyecto, Vista, Imagen, Solución y Ayuda. Por supuesto chequean a ver si todas ofrecen lo mismo que el módulo anterior y se dan cuenta que todas son similares excepto por Solución la cual es muy diferente. Se sienten más familiarizados con el módulo ya que trabajaron con el anterior. Lo primero que hacen es crear un nuevo proyecto para eso selecciona la opción Nuevo en Proyecto pero nota que esta vez además de solicitarle un nombre para el proyecto y una ruta del sistema de archivos para ubicarlo como el modulo anterior, le solicita una capa de centroides y una capa de conexiones. De todas manera se dan cuenta que las capas que

ellos deben ingresar para llevar a cabo su objetivo son las capas resultado del trabajo realizado en el módulo anterior, “Proyecto_Rivera_Centroides.shp” y “Proyecto_Rivera_Conexiones.shp”. Ingresan todos los datos aceptan y se visualizan las capas generadas en el otro módulo. Sintiendo bien encaminados, ahora deciden sin más preámbulos generar los recorridos con los algoritmos. Lo primero que hacen es ir a Solución, y ven la opción Generar Recorridos, la seleccionan y se despliega una pantalla que les pide que seleccionen un algoritmo, MISTERIOSAMENTE ven que dentro de las opciones se encuentran, “Algoritmo Marita” y “Algoritmo Antonio”. Sin cuestionarse demasiado como llegaron los algoritmos a las opciones (por la ansiedad), deciden continuar. Marita escoge cara, Antonio arroja la moneda. Sale cara, comienza Marita. Seleccionan entonces “Algoritmo Marita” y se despliega otra pantalla, esta vez solicitando valores para los parámetros del algoritmo, Marita conociendo su algoritmo, completa los campos y acepta. Luego de unos minutos visualizan líneas, de distinto color al de las conexiones, y que unen distintos centroides, son los recorridos en el mapa. Deciden entonces salvar los recorridos para eso seleccionan la opción Recorridos, Guardar, se despliega una ventana que pide el nombre de los recorridos, Marita ingresa entonces el nombre “RecorridosMarita” y acepta. Llego el turno de Antonio, selecciona Solución, Generar Recorridos, acepta luego en la opción Seleccionar Algoritmo indica la opción “Algoritmo Antonio” y acepta. Se despliega la pantalla de los parámetros del algoritmo Antonio, completa los campos y acepta nuevamente. Luego de unos minutos se ven los recorridos en el mapa. Igual que Marita, Antonio guarda los recorridos como “RecorridosAntonio”. Se dan cuenta que ambos recorridos tienen el mismo color y eso confunde las cosas entonces deciden cambiar el color de los recorridos de Antonio. Pronto, llega el momento de la verdad.

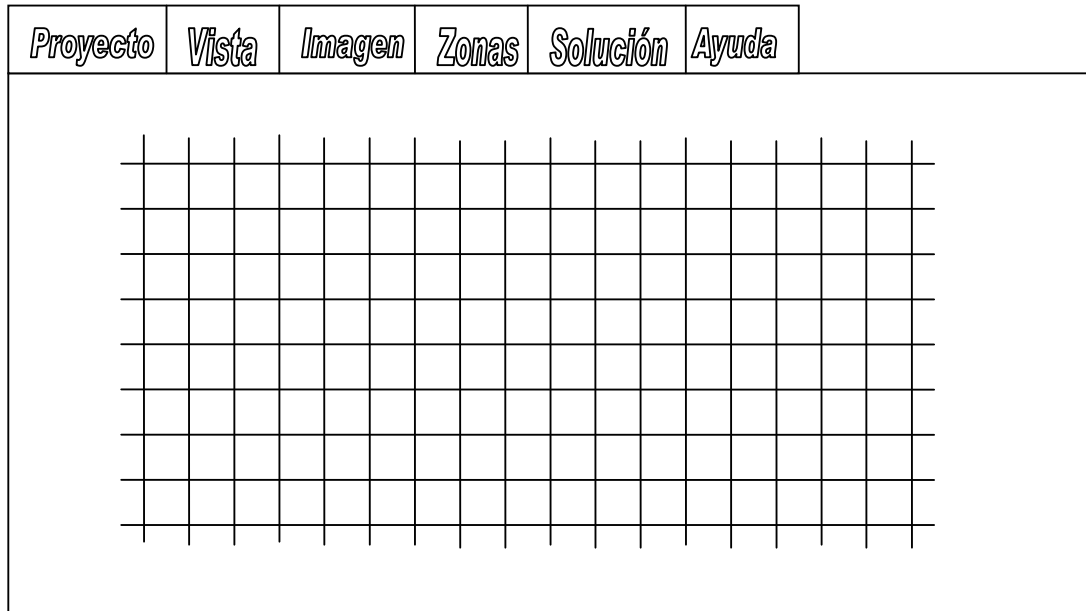


Figura 10.2.1

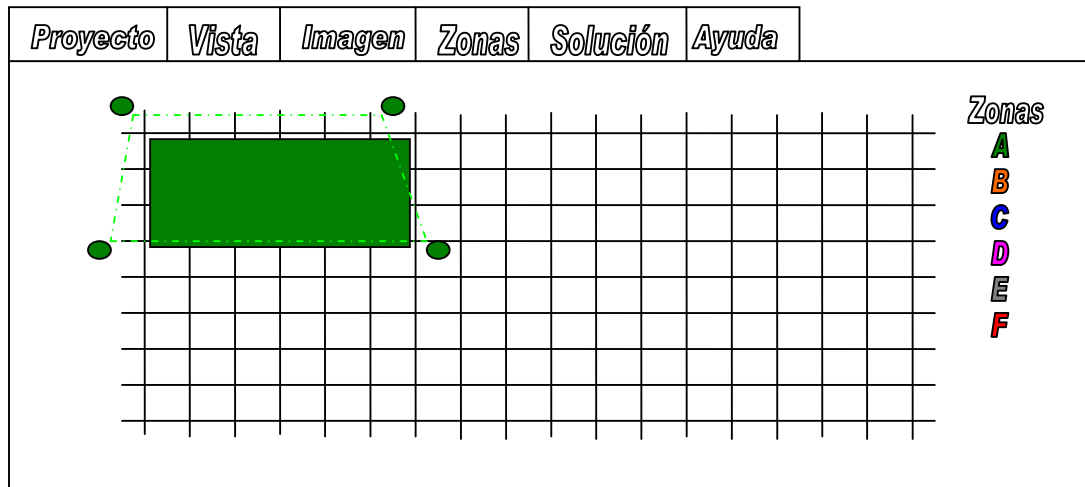


Figura 10.2.2

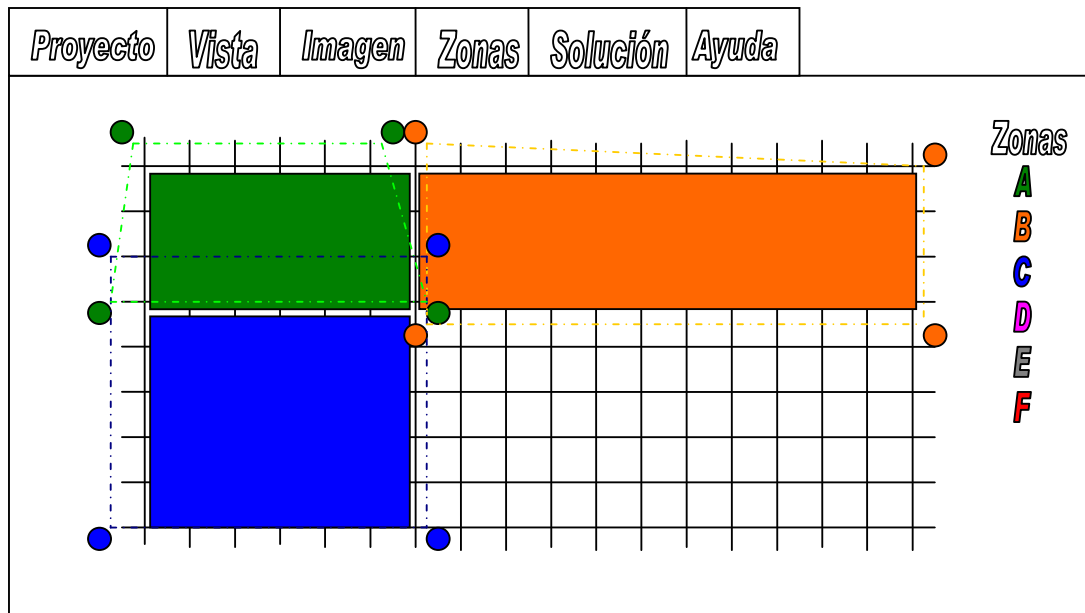


Figura 10.2.3

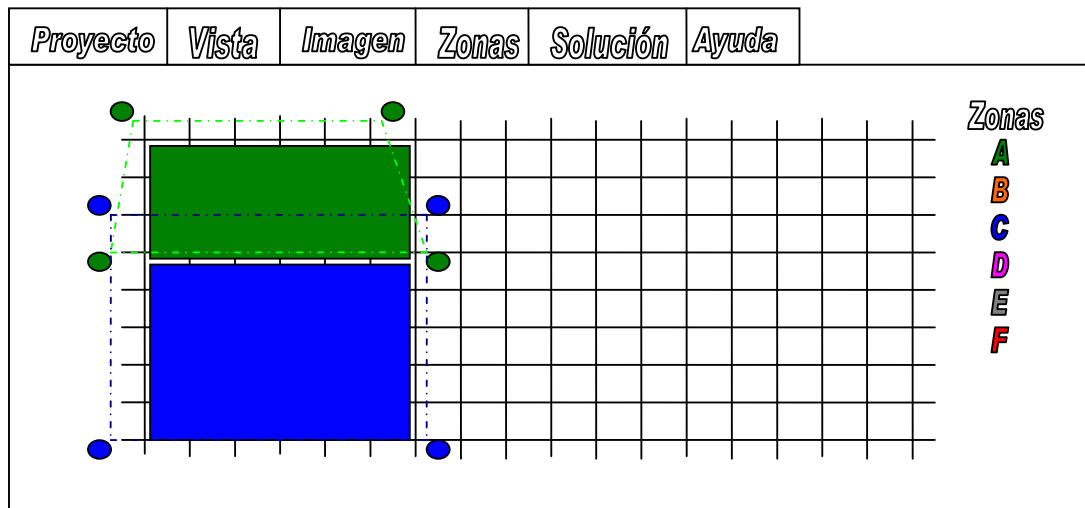


Figura 10.2.4

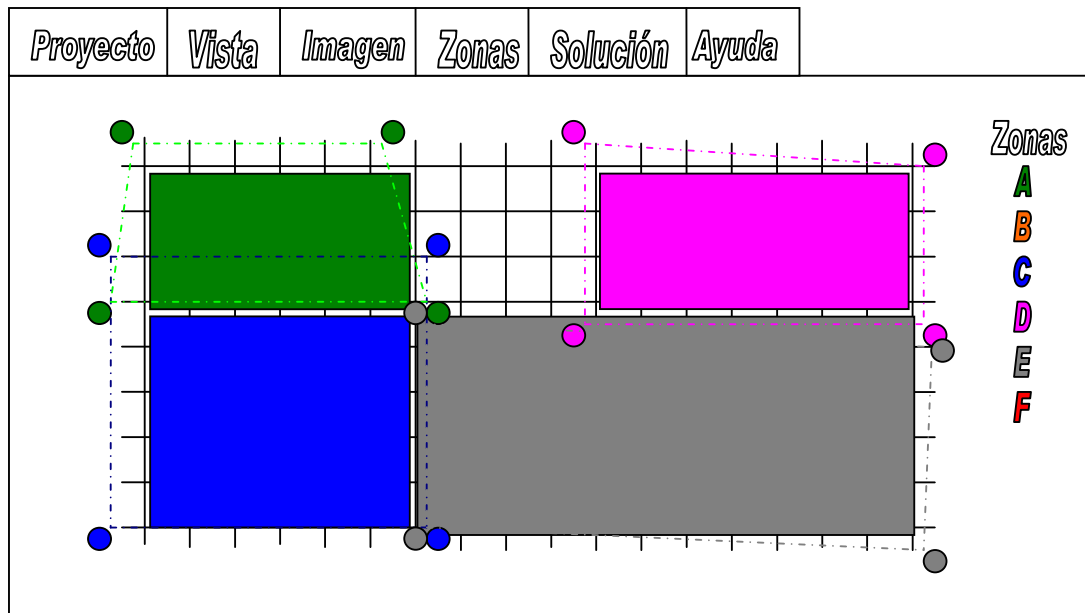


Figura 10.2.5

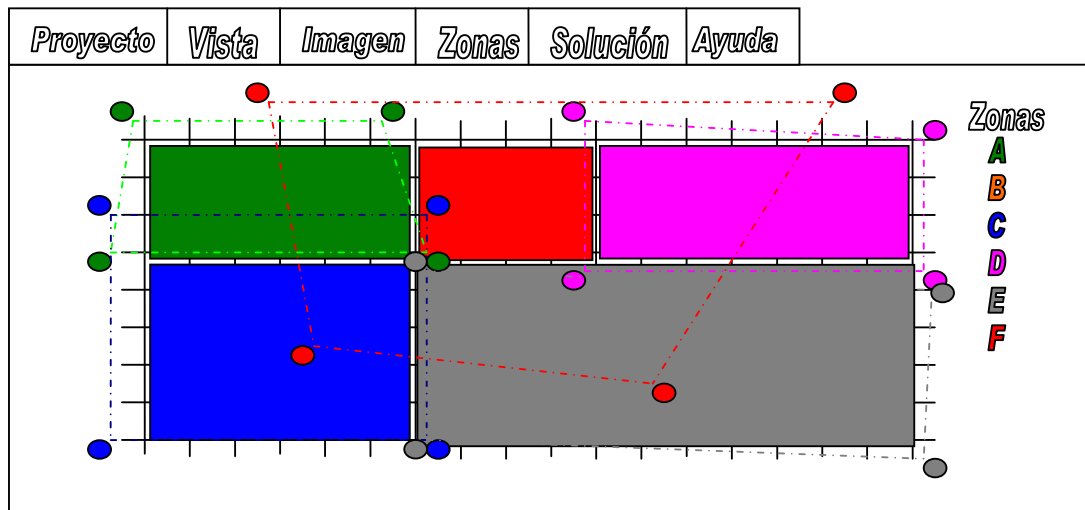


Figura 10.2.6

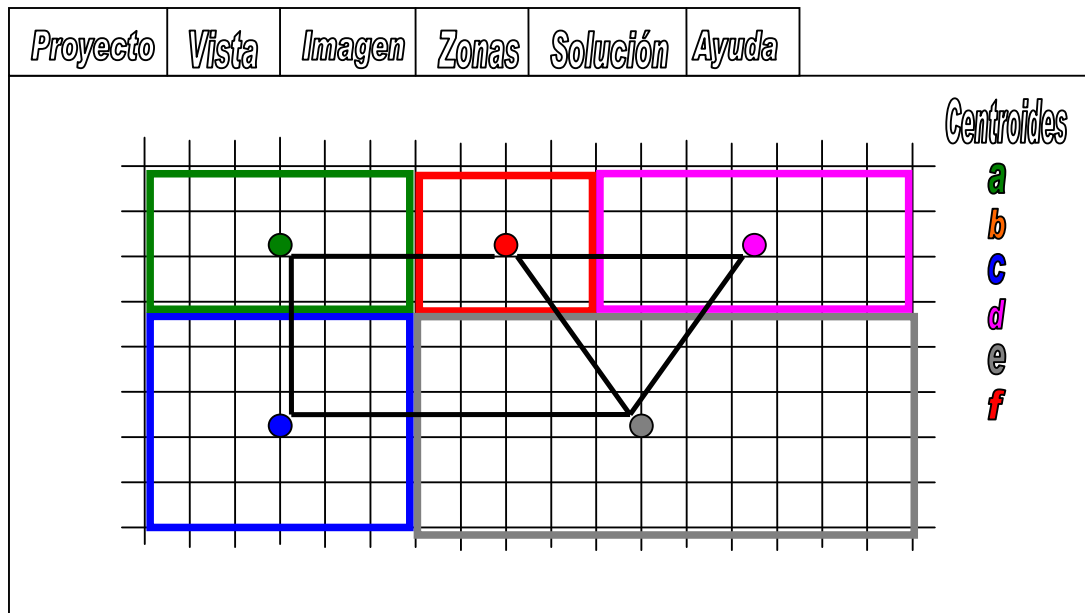


Figura 10.2.7

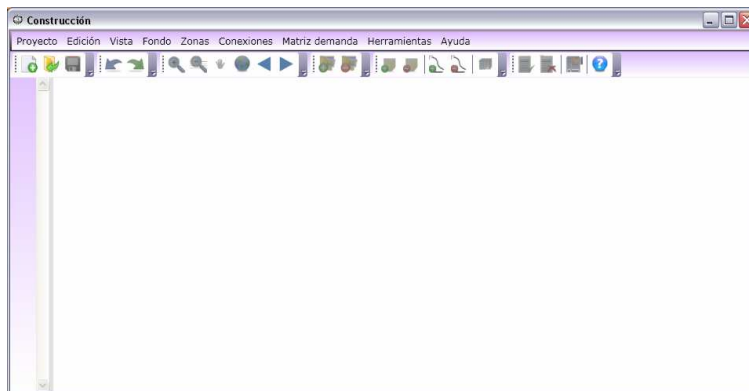
A continuación se muestran algunos prototipos desarrollados en máquina. Los cuales se fueron aproximando más a la realidad pretendida por el usuario.



Prototipo en máquina 1



Prototipo en máquina 2.



Prototipo en máquina 3.

11. Anexo. Actas

En este anexo simplemente se adjuntan algunas actas tomadas durante las reuniones que mantuvimos con los tutores, mediante las cuales se fueron validando los diversos requisitos que debía cumplir el sistema.

Fecha: 12 de junio de 2008

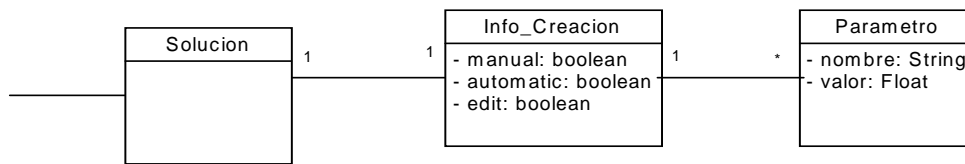
- *Documento:*

Este documento es el acta de reunión con los tutores del proyecto llevada a cabo el día 11 de junio de 2008 a las 15 horas en las oficinas del Departamento de Investigación Operativa, Instituto de Computación. La reunión duro aproximadamente una hora.

- ***Temas Tratados:***

- ***Evaluación de la documentación presentada hasta el momento:***
- Las funcionalidades básicas se encuentran especificadas de acuerdo a la percepción del cliente en el documento de requisitos (Anexo Análisis y Diseño). A lo largo del proyecto se irán afinando los mismos.
- Algunas observaciones en cuanto al documento de requisitos, las cuales deben ser consideradas en el mismo:
 - Referenciar secciones en forma más concreta (título, sección, etc.).
 - las conexiones se agregan “manualmente”. En una primera instancia es suficiente que la creación de conexiones entre centroides, sea de forma manual. Aunque también se manejó la posibilidad de aprovechar la herramienta SIG para facilitarle al usuario la construcción de estas conexiones, brindando una generación automática y que luego el propio usuario pueda editarlas. Por ejemplo que se generen las conexiones entre los centroides adyacentes.
 - “arcos y nodos” vs. “aristas y vértices”. Se definió como correcta la utilización de aristas y vértices para el caso estudio, ya que las aristas no tienen dirección, tal como se maneja en el problema planteado.
 - Puntos de demanda: se considera que la explicación es correcta y necesaria (manzanas). Sirve para conteos tanto de hogares como de paradas (genérico).
 - Se cree necesaria expresar la demanda en función de la matriz origen – destino. La demanda intrazonal es nula.
 - a modo de encapsular mejor las funcionalidades de herramienta que resolverá el problema, se limitará la cantidad de capas a ser

- ingresadas. Capas de: calles (polilíneas), cruces (puntos), puntos demanda (puntos), zonas (polígonos), centroides (puntos) y conexiones entre centroides (polilíneas). Se busca independizar las funcionalidades necesarias con las que puedan ser hechas en cualquier herramienta SIG.
- También se pretende fijar los campos (atributos) que serán manejados en las diversas capas, como conjuntos acotados. Es decir las capas deben contener como mínimo esos atributos, pero los adicionales no serán considerados.
 - Faltaría refinar el requisito de “Editar solución”. Se aconseja la utilización del proyecto de grado del 2003.
 - Se pretende independizar la aplicación de los parámetros que requiera el algoritmo de optimización. Por ejemplo la ventana que solicite dichos parámetros debe variar de acuerdo a los que el algoritmo necesite.
- Algunas observaciones en cuanto al documento del modelo de dominio (Anexo Análisis y Diseño), las cuales deben ser consideradas en el mismo:
- Habría que clarificar mejor la información de entrada y de salida requerida por los 2 módulos (preparación del caso de estudio y manipulación del mismo).
 - Concepto de recorrido: se puede definir como secuencia de aristas de ida. (la vuelta será el conjunto inverso).
 - Concepto de solución: conjunto de recorridos sobre un escenario. El escenario se compone de una red y matriz (caso de estudio). Se pueden generar varias soluciones sobre un escenario. Se plantea la idea de guardar un registro sobre la información de creación de la solución. Con datos que incluyan si la generación fue automática, manual, o automática y editada manualmente. Si lo llevamos a un modelo de dominio, esta sección se vería algo así:



- Habrá que desarrollar el concepto de “proyecto” dentro de la aplicación para poder luego establecer su forma de uso. Es decir si cada proyecto incluye un escenario y las diversas soluciones generadas, podría ser suficiente manejar un proyecto por vez (simultáneamente).
- El concepto de “Flota” heredado de proyectos anteriores se considera innecesario, puesto que puede ser representado como un parámetro más, que significa la capacidad del ómnibus.
- Del modelo de dominio se puede quitar los conceptos de flota y bus.

- **Problema de zonificación y las distintas opciones:**
 - Como mínimo se debe brindar la posibilidad que el usuario determine las zonas en forma manual.
 - También se vio útil darle la posibilidad de generación automática de las zonas por parte de la herramienta.
 - Investigar las facilidades que proporciona la herramienta SIG para la generación automática de zonas. (Por ejemplo cada 400 metros, o por cantidad de la demanda)
 - La aplicación puede ser consultada para obtener la información acerca de las demandas de las zonas (o manzanas). Podría ser mediante la coloración de las zonas, de acuerdo a alguna referencia. Ver funcionalidades brindadas por la herramienta SIG.

- **Estudio de las diversas tecnologías:**
 - MapWindow, GRASS, gvSIG, JUMP, OrbisCad, FDO, GeoTools.

- Decisión basada en: Funcionalidades brindadas, funcionalidades necesarias y ambiente de trabajo.
- También considerar para la decisión: soporte que tenga la herramienta, continuidad (cuando salió la última versión), usabilidad.
- Marcar las características necesarias que debe cumplir la herramienta SIG, tomar como referencia el libro GIS – Transportation...
- Considerar los tutoriales o manuales que sirvieron guía, indicando nuestra opinión sobre los mismos.
 - o **Planificación para las próximas semanas:**
- La próxima reunión será posterior a la visita de los docentes y Manuel a la ciudad de Rivera, donde se continuará ajustando el proyecto a las necesidades de los futuros usuario.
- Visita se llevará a cabo en la semana 22-29/06.

La fecha será marcada más adelante para martes o jueves posterior a la visita.

Fecha: 03 de Julio de 2008

Versión: 1.0

- **Documento:**

Este documento es el acta de reunión del proyecto con el tutor Antonio Mauttone, llevada a cabo el día 03 de julio de 2008 a las 15 horas en las oficinas del Departamento de Investigación Operativa, Instituto de Computación. La reunión duró aproximadamente una hora.

- **Temas Tratados:**

o **Tecnologías**

Se planteó por parte del grupo la voluntad de determinar la tecnología GIS, que se utilizará para la implementación del proyecto. Por los avances conseguidos hasta el

momento, la decisión se reduce a los últimos ensayos entre las opciones de MapWindows o GeoTools. Con las ventajas y desventajas que plantea cada una, las cuales serán detalladas en el documento de tecnología.

○ ***Validación de Interfaces.***

Se presentó en forma impresa un primer prototipo en formato de esquema, para dar al usuario la posibilidad de ir validando las distintas interfaces para los 2 módulos del proyecto. A continuación se puntuarán ciertas observaciones:

Módulo 1 – preparación caso de estudio.

- Zonas. Crear Zona: al momento de crear un nuevo proyecto (en el que aún no existe zona) se planteó.
 - Asumir que se maneja una capa de zonas.
 - Se genera una nueva capa al crearse un nuevo proyecto.
- Zonas. Agregar Zonas: En una primera etapa las zonas serán realizadas por el usuario en forma de polígonos simples, con alguna restricción del tipo que no hayan islas o superposición de zonas. El ajuste de zonas a calles más cercanas o autogeneración de zonas en base a algún criterio será para etapas posteriores.
- Solución. Generar grafo: Dividir la generación del grafo en forma manual o automática.
- Solución. Agregar Conexiones: Pensar los modos de edición, por ejemplo la selección impar del botón izquierdo del ratón (“click”) será nodo origen y la selección par el nodo destino. Existiendo la posibilidad de finalizar con algún botón disponible en pantalla. Seguir ejemplo del proyecto de grado del año 2003.
- Ya que el primer módulo no parece ser complejo a nivel algorítmico se recomienda que sea sencillo en cuanto a la facilidad de uso.

Módulo 2 – Manipulación del caso de estudio.

- ver el tema del manejo de imágenes georeferenciadas. Distribución de píxeles en base a coordenadas. Archivos .tiff y .shp. Otra posibilidad es procesar las imágenes con la herramienta ArcGIS previamente para obtener los archivos deseados.

- Cada recorrido será una capa distinta para poder activarlo individualmente.
- La activación de cada recorrido se puede diferenciar cambiando el color y aumentando el ancho de las líneas.
- Revisar el proyecto de grado de 2003 para definir la visualización de soluciones, para que sea una por vez. Cada solución es un conjunto de recorridos, por lo tanto si se abrieran varias al mismo tiempo sería bastante confuso para el usuario. Tener en cuenta que cada corrida del algoritmo para el caso de estudio (ciudad de Rivera) genera alrededor de 200 soluciones. Ver la posibilidad para guardarlas en una carpeta de cada proyecto para después poder visualizar alguna.
- Definir archivo o archivos donde el algoritmo escribe las soluciones. (formato XML o txt)
- Posibilidades para visualizar la solución, 2 extremos:
 - o Mínimo para poder visualizar la solución – lectura de archivo
 - o U opción de manejar toda la información con la GEODatabase.
- El algoritmo que genera la solución **siempre** parte de una vacía.
- Debe existir otra posibilidad de crear soluciones manuales a partir de una vacía.
- Tomar ejemplo del proyecto de grado 2003 para la manipulación de recorridos y escalas. Quizás lo más sencillo sea tomar la escala del .shp.
- Manipulación de base de datos. Opción Access. En Java- JDBC o Visual Basic es directo.
 - o ***Link propuesto por el tutor***
- Software de Citilabs, programa Viper.
- Sin mayor importancia. Sería un ejemplo de una interacción con el usuario poco intuitiva.
 - o ***Planificación para las próximas semanas:***
- La próxima reunión será el 16/07
- Afinar documentación y cuadro tecnológías.

Fecha: 16 de Julio de 2008

- ***Documento:***

Este documento es el acta de reunión del proyecto con el tutor Antonio Mauttone, llevada a cabo el día 16 de julio de 2008 a las 15 horas en las oficinas del Departamento de Investigación Operativa, Instituto de Computación. La reunión duró aproximadamente una hora.

- ***Temas Tratados:***

o ***Capa de fondo.***

Posibilidades: Archivos .gif .jpg .bmp., .tiff (Imágenes georeferenciadas), .dwg (autocad), shp (ESRI shapefile, poligonos), grillas (ESRI grid).

La definición exacta de los tipos de archivos que serán aceptados como capa de fondo se definirá con más precisión en etapas futuras. En principio a esta capa se la considera sólo de lectura.

o ***Base de datos ACCESS o similar en OpenOffice.***

La base de datos donde se almacenará la información correspondiente a la ciudad de Rivera, se hará con la herramienta Microsoft Access. Se considerará el futuro estudio de compatibilidad con el programa análogo pero de OpenOffice.

o ***Tecnología: MapWindows***

Finalmente se optó por la herramienta de SIG: MapWindows. Los motivos de la elección se detallarán en el documento de análisis de tecnologías. Quizás lo más destacable de esta herramienta es la facilidad de instalación y creación del ambiente de desarrollo comparada con el resto de herramientas de no propietarias.

o ***Entorno de Programación: Visual Studio 2008/2005***

Se decidió utilizar la plataforma de desarrollo Visual Studio 2008.

Primero por la compatibilidad de lenguajes que proporciona con respecto a la herramienta SIG. Segundo porque es positivo adaptarse a las últimas tecnologías existentes al momento de implementación, brindando esto una mayor perspectiva a planes futuros de continuidad en el proyecto.

Se aprovechará las licencias Microsoft, que cuenta la Facultad de Ingeniería de la UdelaR para la instalación del entorno de desarrollo.

○ ***Lenguaje: C#, VB. Preferencia? Desventajas .NET de otros años.***

El problema que se presentó en el desarrollo de la herramienta en proyectos anteriores fue la dificultad existente con la mezcla de distintos lenguajes para la implementación.

El grupo ha decidido llevarla a cabo en C# por la similitud que tiene este lenguaje con los vistos durante la carrera de grado, como ser Java.

○ ***¿2 aplicaciones?- separadas***

Se definió que en un principio se implementarán 2 aplicaciones:

- Preparación del Caso de Estudio.
- Manipulación del Caso de Estudio.

De esta forma se logra independencia de ambos módulos y se mantiene la posibilidad de trabajar concurrentemente sobre ellos.

○ ***Algoritmos de optimización.***

Los algoritmos serán programas (ejecutables) independientes a las aplicaciones que resuelve el proyecto.

La comunicación entre los algoritmos y la aplicación de “Manipulación del Caso de Estudio” existirá mediante la lectura y escritura de archivos .txt o .xml

○ ***Planificación para las próximas semanas:***

- La próxima reunión será el 30/07 a las 15 hs. Enviar recordatorio.
- La idea es arrancar con la implementación y empezar a presentar esquemas (prototipos elementales) ya computarizados
- Pedido de instalación de herramientas para el entorno de desarrollo en aunque sea 2 máquinas del Depto. de Investigación Operativa.

12. Anexo. Transporte Público

Esta sección describe las etapas y los subproblemas en la planificación del transporte público, esta basada en [DH 2003] y [ORW 1994].

12.1 Planeamiento Estratégico

Al nivel estratégico, el planeamiento del transporte público se encarga del diseño de las rutas de transporte público y la red. Esto implica diseñar una red de rutas que enfrente la demanda de pasajeros. Ya que la demanda está basada en gran medida en el diseño de la red, el problema de diseñar la red recae sobre métodos para determinar las rutas escogidas por los pasajeros (o la asignación de rutas) que sirve para sus orígenes y destinos.

12.2 Diseño de la red

El problema de diseño de una red de transporte público es de alguna forma más complicado que el problema tradicional de diseño de una red. Además de determinar que aristas van a ser incluidas en la red, el diseño de una red de transporte público incluye componer estas aristas en rutas fijas, y determinar la frecuencia del servicio en cada ruta. El resultado del diseño de la red, entonces, debe incluir un conjunto de rutas y sus frecuencias. Más comúnmente, el problema es formulado sobre un grafo con nodos, aristas, y (subsecuentemente) rutas. Sea $G=(N,A)$ un grafo con N el conjunto de nodos y A el conjunto de aristas, y R el conjunto de rutas. Los nodos representan intersecciones (por ejemplo cruce de calles), pero también pueden representar centroides de zonas donde una zona geográfica es representada por un único punto (el centroide). Una arista entre nodos representa un modo particular de transporte entre nodos, y una ruta representa una secuencia de nodos y aristas de un modo único.

Como entrada al problema, la formulación típicamente asume que existe una matriz origen destino (O-D), que representa la demanda entre un conjunto de nodos o zonas, sobre una base diaria o para un período específico del día. Alternativamente, es razonable (aunque complicado) asumir que la demanda es endógena y determinada como un problema de equilibrio, en donde los flujos son una función del diseño de la

red. Con los flujos origen destino y una asignación de estos flujos sobre la red, el conjunto de rutas y sus frecuencias debe ser determinada.

Uno de los desafíos del diseño de redes es la especificación de la función objetivo. Lo más común es minimizar el tiempo total del viaje o el costo general del viaje. El costo general puede ser hallado aplicando diferentes pesos en la función objetivo a los diferentes componentes de tiempo del viaje como tiempo a pie (o tiempo de acceso), tiempo inicial de espera, tiempo en el vehículo, tiempo de transferencia, y tiempo de egreso separadamente. Algunas formulaciones también incluyen el número de transbordos como un componente en el costo general.

Además, el costo de los operadores también debe ser considerado, ya sea explícitamente en la función objetivo o a través de restricciones en el presupuesto total (o ganancias o pérdidas en operación). Estos costos pueden incluir el costo de operación, dado como una función de distancia de la ruta (en distancia o tiempo) y frecuencia, y el costo fijo de la flota de buses y/o infraestructura a lo largo de la ruta de la red (para redes de tránsito férreo). Si los costos del operador están incluidos, un objetivo compuesto debe ser formulado, o el problema puede ser especificado como un problema de programación multiobjetivo.

Además de esta formulación tradicional del objetivo, diversas otras restricciones usualmente entran en el problema en la práctica; esto incluye: (1) asegurar una adecuada cobertura en la red para proveer acceso a nodos o zonas específicas en el área de servicio; (2) asegurar un mínimo de frecuencias en los servicio a nodos o aristas específicas de la red; y, (3) cualquier otra consideración de diseño como la disponibilidad de infraestructura o prioridad para rutas.

En la práctica, el problema de diseño de redes incorpora cada uno de estos objetivos y restricciones a través de una formulación del problema más interactiva, donde rutas de muestra son construidas por métodos computacionales, pero al final son seleccionadas de forma manual en conjunto por diseñadores de redes y planificadores.

El problema de diseño de redes es NP-difícil. Como resultado, el problema se enfrenta utilizando técnicas heurísticas que resuelven problemas de tamaños considerables.

12.2.1 Asignación de pasajeros

Una de las cuestiones críticas en el planeamiento estratégico es determinar la demanda en cada ruta y otras medidas de consumo de servicios. Dos de las medidas estratégicas más comunes para medir el rendimiento, desde el punto de vista del pasajero, se relaciona con la cantidad de tiempo y dinero gastado en viajar sobre la red; elementos del camino del pasajero desde el origen al destino. Por lo tanto, la asignación de pasajeros es crítica para determinar el rendimiento del sistema.

El problema de la asignación de pasajeros puede ser definido como sigue. Dado un flujo origen destino, ¿cuales son los flujos o caminos a través de la red de transporte público, escogidas por el pasajero? En la formulación de este problema, el objetivo de los pasajeros supone minimizar el tiempo de viaje o el costo general. El tiempo de viaje puede consistir en algunas o todas las siguientes variables, con quizás diferentes pesos; el tiempo de acceso a la parada, el tiempo en la parada, el tiempo en el vehículo, el tiempo de transbordo, el número de transbordos, el tiempo de egreso, y el costo monetario. El pasajero entonces se enfrenta a la tarea de seleccionar una ruta de un conjunto que lo pueden llevar del origen al destino con mínimo tiempo o costo general. En la literatura, este problema es tratado dentro del problema de diseño de redes, como parte de la tarea de determinar las frecuencias de las rutas, y también es tratado como un problema particular.

En contraste a la simplicidad de la definición del problema, hay un número de aspectos del mismo que han llevado a diferentes enfoques de la investigación. Un concepto importante en la asignación de pasajeros es la determinación del camino de “costo mínimo”. Se incluyen importantes elementos en la definición de costo:

1. la caracterización de la dependencia-tiempo y atributos estocásticos en el camino de costo mínimo.
2. la caracterización de la solución como: (1) un *único camino*, incluyendo sólo una ruta o una combinación de rutas; (2) un camino que incluye un conjunto de *líneas comunes*, incluyendo casos donde varias rutas se pueden superponer en alguna parte del camino más corto; o, (3) una estrategia, permitiendo a los pasajeros escoger su propia regla de embarque mientras viajan desde el origen al destino.
3. el efecto de capacidad y multitud en la red de transporte público.

12.3 Planeamiento Táctico

El planeamiento táctico, se encarga de los pasos intermedios en el proceso de planificación en donde la frecuencia de las rutas es construida y el programa del servicio es determinado. Se incluyen estas dos partes juntas en el nivel táctico porque son predominantemente orientadas hacia la estructuración y mejora del servicio de pasajeros.

12.3.1 Ajuste de frecuencias

El proceso de determinar la frecuencia para las rutas de transporte público ya ha sido introducido en el proceso de diseño de redes. Mientras que un conjunto de frecuencias son un producto necesario del diseño de redes, es también cierto que las agencias de transporte público evalúan y determinan frecuencias sobre las rutas más a menudo. Variaciones en los patrones de la demanda de pasajeros y pequeños cambios en el diseño de rutas pueden precipitar la necesidad de un ajuste de frecuencias.

El problema del ajuste de frecuencias puede ser encarado de diferentes maneras. El objetivo principal es seleccionar frecuencias que maximicen el servicio de pasajeros, el cual puede estar definido de diferentes maneras, sujeto a un número posible de restricciones. Estos incluyen restricciones sobre el tamaño total de la flota (el cual puede asumirse fijo para este proceso), una restricción de que la capacidad de la ruta debe ser suficiente para la demanda, y cualquier restricción política sobre la mínima frecuencia deseable. Otra información de entrada incluye el tiempo en cada ruta de ida y vuelta y descanso requerido. Con esta información, la agencia de transporte público asigna la flota a las rutas; esta asignación indica directamente la frecuencia del servicio en cada ruta. Finalmente, estas frecuencias deben ser especificadas para un horario en el día y un día en la semana.

12.3.2 Construcción de la tabla de horarios

La construcción de la tabla de horarios es el proceso que convierte la frecuencia de servicio de cada ruta fija en una tabla de horarios. La entrada a este proceso incluye la estructura de las rutas, incluyendo tiempo de viaje entre los principales puntos en el tiempo, la frecuencia de los servicios, y cualquier tiempo de descanso necesario en las terminales u holgura en el programa (tiempo extra construido en el programa) entre los

principales puntos en el tiempo del servicio. El resultado es un conjunto de viajes y de tiempos programados en las terminales y en los principales puntos en el tiempo de las rutas.

En los principales métodos de planeamiento de transporte público, la construcción de la tabla de horarios se incluye dentro del planeamiento operacional, dado que ocurre con frecuencia, con cada ajuste de servicio (por ejemplo cada 3-6 meses). También, es a partir de las tablas de horarios que se construyen los programas de vehículos y tripulación. Se incluye como un elemento del planeamiento táctico dado que involucra la determinación del programa para maximizar el servicio a los pasajeros. Esto en contraste con los problemas de programación de vehículos y tripulación que típicamente están asociados a la planeación operacional, que se enfoca en minimizar los costos operativos del transporte público.

En muchos casos, la construcción de la tabla de horarios es directa, principalmente asumiendo que el intervalo de tiempo entre los vehículos de una misma ruta es constante y que la demanda es relativamente uniforme sobre el período de tiempo de interés. Se especifican algunos tiempos de reloj para el primer viaje de vehículo de cada período, y las salidas de vehículos de la terminal o puntos de máxima demanda son múltiplos del intervalo de tiempo entre vehículos de la ruta. Tiempos estimados entre puntos en el tiempo son usados para determinar el programa, y los tiempos de descanso son usados para programar los viajes de retorno. La simplicidad de esta tarea puede explicar en parte por que ha recibido relativamente poca atención de parte de los investigadores. Sin embargo, una complicación mayor en la construcción de la tabla de horario sucede cuando los horarios intentan estar coordinados en las paradas de transbordo o terminales.

Quizás el desafío más apremiante en la construcción de las tablas de horarios es la sincronización de los horarios de los vehículos de manera que los transbordos en la red sean en tiempos razonables. Específicamente, a uno le gustaría coordinar el tiempo de arribo de un vehículo en una ruta con otro en otra ruta para que la transferencia de pasajeros entre las rutas pueda realizar la conexión en el menor tiempo posible. Mucho de los primeros trabajos sobre este problema se enfocaban en métodos para sincronizar en puntos específicos en el tiempo; métodos más recientes usan heurísticas para

problemas de redes grandes. La naturaleza combinatoria del problema indica que es NP-difícil.

12.4 Planeamiento operacional

Dado un conjunto de tabla de horarios de viajes para operar, un conjunto de recursos habilitados (buses y tripulación) y la distribución de los recursos en los distintos depósitos del agente, la fase de planeamiento operacional se encarga de construir los programas de vehículos y tripulación que minimizan el costo total respetando todas las restricciones operacionales y regulaciones laborales. Esta fase también incluye el planeamiento de las asignaciones de buses a los espacios de estacionamiento en los depósitos y su despacho, así como de establecer el programa de mantenimiento.

12.4.1 Programación de vehículos

La programación de vehículos juega un rol importante en el manejo del transporte público por parte de las agencias ya que es el primer paso de planeamiento donde el foco principal es puesto en la minimización de costos, llamémosle, costos de adquisición y operación de los buses. En los pasos anteriores el énfasis estaba puesto en el servicio a los pasajeros, el cual es fijo en la programación de vehículos. Es más, en esta etapa, el servicio ofrecido a los clientes está completamente determinado por las tablas de horarios de los viajes fijas ahora. En general, la misma tabla de horarios para el día se aplica a todos los días de la semana excepto los fines de semana donde se utiliza una tabla de horarios diferente. Estas tablas de horarios son usualmente válidas por un cierto período de tiempo (temporada). Por esto la programación de vehículos en el nivel de planeamiento necesita ser llevada a cabo una vez para cada tabla de horarios y temporada. Es de notar que en las ciudades grandes los buses operan las 24 horas del día, lo que hace más difícil de definir un problema diario. Sin embargo, dada la poca actividad durante la noche, en la práctica una tabla de horarios de 24 horas es dividida en una tabla para el día y otra para la noche.

El problema de programación de vehículos enfrentado por las agencias de transporte público corresponde al problema de programación de vehículos con un depósito (SDVSP, single-depot vehicle scheduling problem) cuando la agencia opera su flota de buses utilizando un único depósito, y al problema de programación de vehículos con

múltiples depósitos (MDVSP, multiple-depot vehicle scheduling problem) cuando se utilizan varios depósitos. Este problema puede ser planteado como sigue. Sea $T=\{1,2,\dots,n\}$ un conjunto de n tablas de horarios de viajes donde el viaje i perteneciente a T comienza en el tiempo si y finaliza en el tiempo ei . Estos viajes son calificados como activos ya que los pasajeros viajan en ellos. tij denota el tiempo de viaje (posiblemente incluyendo tiempo de espera) entre el lugar de fin del viaje i y el lugar de comienzo del viaje j . Asumimos que este tiempo de viaje es el mismo para todos los vehículos. Dos viajes i y j se dice son compatibles si y sólo si pueden ser cubiertos por el mismo vehículo (j inmediatamente seguido a i), esto es, si y sólo si $ei + tij \leq sj$. El viaje entre ellos es llamado viaje expreso ya que no hay pasajeros a bordo.

Sea $K=\{n+1, n+2, \dots, n+m\}$ el conjunto de m depósitos de buses que deben ser asignados para cubrir los viajes activos. El depósito k perteneciente a K maneja vk buses idénticos los cuales deben comenzar y finalizar su programa en este depósito. Un bus partiendo de un depósito para alcanzar su punto de partida se dice que realiza un viaje pull-out, mientras que realiza un viaje pull-in cuando retorna al depósito desde el punto final de un viaje activo. Un programa factible para un bus en un depósito k se compone de un viaje pull-out comenzando en k , una secuencia de viajes activos separados por viajes expresos, y un viaje pull-in finalizando en k . Viajes activos consecutivos deben ser compatibles.

La estructura del costo es como sigue. Se incurre en un costo cada vez que un vehículo realiza un viaje expreso, pull-out o pull-in. Este costo es denotado por cij para cada viaje expreso que conecta los viajes i a j , por ckj para los viajes pull-out que conectan al depósito k con el punto de partida del viaje j , y cik para los viajes pull-in retornando al depósito k desde el punto final del viaje i . Es de notar que los viajes activos no llevan costo ya que representan una cantidad fija para cualquier solución factible. Es de notar también que los costos fijos de los vehículos pueden ser sumados a los costos de los viajes pull-out y pull-in. El costo de un programa es la suma de los costos de los viajes que contiene.

El MDVSP puede ser definido como el problema de hallar un conjunto de programas de vehículos tal que cada viaje activo i perteneciente a T es cubierto por exactamente un programa, y como mucho vk programas son definidos para cada depósito k perteneciente a K , y la suma de los costos de los programas es mínima. El SDVSP simplemente

corresponde al caso donde $|K|=1$. En algunas versiones del MDVSP algunas restricciones adicionales son consideradas. Por ejemplo, pueden existir restricciones de compatibilidad de viajes a los depósitos los cuales restringen el conjunto de depósitos que pueden proveer un vehículo a un viaje activo. Es de notar que, en el contexto del transporte público, un viaje expreso que involucre un gran tiempo de espera antes del comienzo del siguiente viaje activo es usualmente remplazado por un viaje pull-in, una período de espera en el depósito, y un viaje pull-out. Los programas de los vehículos son vistos como secuencias de bloques, donde cada bloque consiste de una secuencia de viajes que comienzan y terminan en el mismo depósito sin retornar en el medio de la secuencia.

El SDVSP aparece para agencias de transporte público pequeñas y medianas que cuentan con un único depósito. También puede aparecer como un subproblema del MDVSP. El SDVSP es resoluble en tiempo polinomial. De hecho, puede ser modelado como un problema de flujo de costo mínimo. También ha sido formulado como un problema de asignación lineal, un problema de transporte, un problema de cuasi asignación.

El MDVSP es común en agencias de transporte público medianas, e inevitable en grandes. Ha sido estudiado por más de 25 años. Es un problema NP-difícil.

12.4.2 Programación de tareas

La programación de tareas, también conocida como programación de conductores, es el segundo paso en el proceso de la planeación operacional para las agencias de transporte público. Como el problema de programación de vehículos, el problema de programación de tareas (DSP, duty scheduling problem) es importante desde el punto de vista económico ya que determina la mayoría de los salarios a pagar a la tripulación. El DSP es separable por depósito y consiste en determinar los días de trabajo (también llamado tarea) de la tripulación de un depósito para cubrir todos los bloques asignados al depósito. Debido a que el cambio de tripulación puede ocurrir en varios puntos dentro del bloque, todos los bloques son divididos en secuencias de segmentos según estos puntos de relevo. A los segmentos consecutivos dentro de un bloque asignados a una misma tripulación se les denomina colectivamente pieza de trabajo. Las tareas son entonces compuestas de piezas de trabajo que usualmente están separadas por

descansos. Distintos tipos de tareas, difieren por ejemplo en el número de piezas de trabajo que pueden contener y sus posibles tiempos de comienzo y duraciones. Como ejemplo, pueden existir tareas que contengan una única pieza de trabajo, y tareas divididas conteniendo dos piezas de trabajo. Las tareas están sujetas a una gran variedad de regulaciones de seguridad y reglas de acuerdos colectivos como extensión de las tareas, máxima duración de una pieza de trabajo, e intervalos de tiempo predefinidos donde se deben realizar los descansos. Estas reglas varían según el tipo de tarea.

En general, el objetivo del DSP es doble y consiste en minimizar primero el número total de tareas y segundo el número total de horas de trabajo. Utilizando costos fijos de tarea y una tarifa horaria para las horas de trabajo, este objetivo es usualmente transformado en uno que minimiza el costo total. Resumiendo, el DSP puede ser establecido como sigue. Dado los segmentos de un conjunto de bloques de vehículos, encontrar un conjunto de tareas válido que cubra todos los segmentos y minimice el costo total. Restricciones adicionales como el límite en el número de tareas de un cierto tipo pueden ser tomadas en cuenta.

12.4.3 Programación integrada de vehículos y tareas

En general, la programación de vehículos es realizada antes de la programación de tareas en el proceso operacional de los agentes de transporte público. Dado que las posibilidades de reemplazo de tripulación son numerosas en la mayoría de los contextos, un programa de tareas eficiente puede ser obtenido a partir de un programa de vehículos casi óptimo para obtener al final en general una solución de buena calidad. Por otro lado, cuando las oportunidades de reemplazo son raras, como es el caso en los sistemas de transporte masivo extra urbano o para el proceso de programación línea por línea, un programa de vehículos muy eficiente puede llevar a un pobre programa de tareas o aún a un DSP no factible. Integrar los programas de vehículos y tareas es indispensable en esta situación, la investigación sobre este tema comenzó a ser abordada recientemente.

El problema de programación integrada de vehículos y tareas (IVDSP) puede ser establecido como sigue. Dado un conjunto de tablas de horarios de viajes y una flota de vehículos asignada a varios depósitos, encontrar bloques de vehículos y tareas de tripulación válidas de costo mínimo tal que cada viaje activo es cubierto por un bloque, cada segmento de viaje activo es cubierto por una tarea, y cada viaje expreso, pull-in, y

pull-out utilizado en la programación de vehículos es cubierto por una tarea. Como en el MDVSP, cada bloque debe comenzar y finalizar en el mismo depósito y, como en el DSP, las tareas cumplen con un conjunto de reglas de trabajo y cada tarea debe ser compuesta de viajes que son cubiertos por buses que parten del mismo depósito. Este último requisito es usualmente obligatorio dado que la tripulación es usualmente asignada a un depósito. También se puede agregar restricciones como disponibilidad de vehículos.

12.4.4 Ordenamiento de tripulación

Dado un conjunto de tareas anónimas definidas en un cierto horizonte horario (típicamente, una semana o un mes) para la tripulación asignada a un depósito particular, el ordenamiento de la tripulación consiste en asignar las tareas a la tripulación disponible para formar su programa de trabajo (llamada roster). Como con las tareas, la validación de los roster está restringida por las regulaciones de seguridad y las reglas de acuerdo colectivo. Por ejemplo, un conductor no puede trabajar más de una cierto número de días consecutivos. En la mayoría de las agencias de transito público de Norte América, la tripulación construye su propio roster según la antigüedad, no dejando lugar para la optimización. Por otro lado, en muchas agencias europeas, el principal objetivo del problema de ordenamiento de tripulación es distribuir uniformemente la carga de trabajo entre la tripulación, produciendo un interesante problema de optimización.

12.4.5 Estacionamiento y despacho

Un problema del planeamiento operacional que ha llamado poco la atención en la literatura es el manejo del área de estacionamiento en los depósitos de vehículos. En ciudades congestionadas, los depósitos están usualmente restringidos en espacio y colmados desde el ocaso hasta el amanecer. Estos además contienen distintos tipos de buses requeridos por rutas particulares. Por lo tanto, cuando un bus de un tipo particular debe salir del depósito en la mañana, puede ser necesario mover otros buses para despejar el camino de salida, generando un retardo. Dos alternativas pueden ser consideradas para evitar los retardos. La primera consiste en siempre asignar un bus para ser accesible directamente para el pull-out de la mañana aunque el tipo de bus no

sea el requerido por el programa de vehículos correspondiente. En este caso, decimos que ocurrió un desajuste. La segunda alternativa es reordenar los buses durante la noche de manera que queden todos propiamente posicionados para el pull-out de la mañana. Un cambio de espacio de estacionamiento entre dos buses es llamado un cruce o una maniobra.

Dada una secuencia de arribos de buses durante la tarde, un conjunto de tabla de horarios de pull-out en la mañana, y el tipo de bus requerido para cada pull-out, el problema de estacionamiento y despacho consiste en estacionar los buses en el depósito cuando arriban y despacharlos a los pull-out tal que el número de desajustes o maniobras es mínimo mientras que se satisfacen las siguientes restricciones. Por razones de seguridad, los buses no tienen permitido ir marcha atrás en el depósito. Por tanto, asumiendo que el depósito esta compuesto de carriles que operan como colas, los buses entran en los carriles en un extremo y salen por el otro. Obviamente, la capacidad de los carriles no debe ser excedida. Finalmente, dado el límite de espacio habilitado para realizar las maniobras, estas solo se permiten entre vehículos del mismo carril.

El problema de estacionamiento y despacho es un problema de planeamiento operacional que usualmente necesita ser resuelto diariamente debido a la alta variabilidad de buses habilitados por tipo.

12.4.6 Programación de mantenimiento

Otra área donde la investigación de operaciones puede resultar útil en el planeamiento de las operaciones del transporte público es en el programa de mantenimiento. Dado que los costos de mantenimiento son una de las categorías de gasto más grande en los sistemas de transporte típico, algunas agencias de transporte están investigando ahora en sistemas de programas de mantenimiento que los ayude a reducir los costos de mantenimiento mientras se mantiene un fiable, seguro, y atractivo sistema de transporte público. Desafortunadamente los sistemas de mantenimiento no son aplicables en todas partes. Cuando los estacionamientos de los depósitos, tienen espacio limitado, la asignación de buses se realiza diariamente de acuerdo a los patrones de estacionamiento y es imposible predecir cuantas millas cada bus va a viajar en los siguientes días. En este caso, los buses son simplemente retirados de la flota cuando necesitan mantenimiento y existen los recursos para el mantenimiento. Estos retiros de último

momento pueden reducir la calidad del servicio. Por otro lado, para depósitos con espacio de estacionamiento donde la gran parte de lugares de estacionar están directamente accesibles a cualquier hora, es deseable asignar buses a programas de vehículos que son válidos para un largo período de tiempo. El razonamiento detrás de esta estrategia es que la tripulación puede ser asignada al mismo bus todos los días con la esperanza de que sean más sensibles a cualquier anomalía mecánica en el bus y reporten cualquier pequeño problema antes de que se transforme en uno grande. En este contexto donde el programa de vehículos está fijo por un largo horizonte de tiempo, un sistema de mantenimiento puede ser elaborado para las actividades del programa de mantenimiento cuando los buses no están en condiciones de estar en servicio, y de tal manera de maximizar la utilización de recursos de mantenimiento.

El problema de programación del mantenimiento puede ser definido como sigue. Dado los programas de operación de los buses, los requisitos de mantenimiento, los recursos de mantenimiento y la disponibilidad de la tripulación, el programa para el mantenimiento de los buses y la asignación de estos a las instalaciones existentes tal que cada bus es mantenido en tiempo, mientras que la cantidad de tiempo que los buses están fuera de servicio es mínima. Los requisitos de mantenimiento, involucran varios tipos de mantenimiento, son expresados en términos de máximo kilometraje o número de días entre dos actividades de mantenimiento consecutivas. Notar que las instalaciones de mantenimiento no pueden ser utilizadas para todo tipo de mantenimiento.

12.5 Controles en tiempo real

En la actualidad, una gran variedad de factores exógenos y endógenos pueden afectar el servicio, como el clima, incidentes, variaciones en las condiciones de tráfico, falla de vehículos. Estos factores pueden degradar el nivel del servicio experimentado por los pasajeros. Generalmente se diferencian las inconvenientes en el servicio en menores y mayores: las inconvenientes menores son aquellos que crean pequeñas perturbaciones en el programa (por ejemplo 5-10 minutos), y los inconvenientes mayores causan grandes cortes en el programa. Esta distinción es hecha para diferenciar las respuestas típicas a estos problemas.

Para enfrentar este desafío, los operadores de transporte público emplean una serie de operaciones técnicas de control. Estas van a depender generalmente en la magnitud de las perturbaciones en el servicio. En un servicio normal con perturbaciones menores en el programa e inconvenientes menores en el servicio, detención de vehículo y señales de transporte público preferenciales son las técnicas más comunes que se utilizan. En la detención, un vehículo puede ser detenido en una parada para mejorar el servicio a los pasajeros. Para las señales de transporte público preferenciales, se les otorga a los vehículos un tratamiento preferencial para que pueda atravesar más rápido un cruce señalizado.

Cuando inconvenientes mayores ocurren, medidas de control más serias deben ser consideradas. Estas pueden incluir obviar paradas en una ruta, viajar expreso parte de la ruta u obviar paradas particulares, para alcanzar el programa. También puede ser útil reposicionar un vehículo en una ruta. En vuelta corta (short-turning), un vehículo en una ruta es vaciado y puesto en servicio viajando en el sentido contrario de la ruta, para acomodar la demanda de pasajeros en ese sentido. Expresos en tiempo real pueden ser empleados para re localizar un vehículo en otra parte de la ruta (o a otra ruta) donde puede haber un servicio más grande. Además, vehículos y tripulación extra puede ser habilitados para ser insertados en los servicios a medida que lo necesitan.

13. Glosario

- ActiveX: Es un COM desarrollado por Microsoft que corre en ambiente Windows. Extraído de [MI 2009]
- Aliasing: Efecto indeseable que se obtiene cuando se traza una línea diagonal sobre un dispositivo hardware a baja resolución. El resultado es que una línea o a menudo una curva aparece escalonada. Extraído de [BB 2007]
- Centroeide: el punto en el centro geométrico de un polígono. Extraído de [G 2005]
- COM: (Modelo de objetos componentes) Interfaz estándar para la composición de software desarrollada por Microsoft. Extraído de [MI 2009]
- Cuadrícula: Conjunto de los cuadrados que resultan de cortarse perpendicularmente dos series de rectas paralelas. Extraído de [RAE 2009]
- Datos Raster: Área espacial dividida en celdas regulares. Una celda es un elemento pixel de un raster, contiene un valor que puede significar muchas cosas como ser reflectancia de la luz de una parte del espectro, un color de una fotografía, atributo temático de un tipo de vegetación, una altura, etc. Extraído de [Z 1999]
- Datos Vectoriales: Representan la geometría de los elementos geográficos en forma precisa y compacta, como un conjunto ordenado de coordenadas y atributos asociados. Extraído de [Z 1999]
- Geodatabase: Es un modelo que permite el almacenamiento físico de la información geográfica, ya sea en archivos dentro de un sistema de ficheros o en una colección de tablas en un Sistema Gestor de Base de Datos. Extraído de [Z 1999]
- Georeferencia: Provee información de localización en términos de proyección de mapa o sistema de referencia. Extraído de [BB 2007]
- GPU: Unidad de procesamiento gráfico. En una tarjeta gráfica equivale al microprocesador del ordenador. Las aceleradoras actuales son casi pequeños ordenadores especializados en el tratamiento de gráficos complejos, por lo que incorporan uno de estos dispositivos. Extraído de [PFC 2008]
- Grids: Cuadrícula.

- Grilla: Cuadrícula.
- Metadata: Información estructurada que describe, explica, localiza, o de alguna manera hace más sencillo conseguir, utilizar, o manejar recursos de información. Extraído de [N 2004]
- Metaheurística: Clase de métodos aproximados, especialmente diseñados para atacar problemas “duros” de optimización combinatorio en los que las heurísticas clásicas han fallado en cuanto a eficiencia y efectividad. Es un proceso iterativo de generación que guía una heurística subordinada, combinando diferentes conceptos “inteligentemente” para explorar y explotar el espacio de búsqueda usando desde estrategias de aprendizaje hasta estructuras de información con el objetivo de encontrar eficientemente una solución “casi” óptima. Extraído de [OK 1996]
- NFS: (Network File System) Es el protocolo de compartimiento de archivos y recursos de las redes Unix. Permite al sistema operativo ver los archivos en las computadoras de la red, como si fueran locales. Extraído de [PC 2008]
- Nomenclátor: Catálogo de nombres, ya de pueblos, ya de sujetos, ya de voces técnicas de una ciencia o facultad. Extraído de [RAE 2009]
- OGC GML: (Geographic Mark-up Language) Es una gramática XML para expresar elementos geográficos. Extraído de [OGC 1994]
- OGC Grid Coverage Service: Define métodos que permiten la interoperabilidad entre las distintas implementaciones de proveedores de datos y de software, proveyendo análisis de grilla (raster) y capacidad de procesamiento.
- OGC FES: (Filter Encoding Standard) Define codificación XML para expresiones filtros. Una expresión filtro combina lógicamente restricciones sobre las propiedades de los elementos para identificar un subconjunto particular de elementos.
- OGC Servicio de catálogo: Brinda la capacidad de publicar y buscar colecciones de información descriptiva (metadata) acerca de información geográfica, servicios y recursos relacionados. Extraído de [OGC 1994]
- OGC SFS: (Simple Feature Standard) Provee una forma común y bien definida para que las aplicaciones almacenen y accedan a los datos de los elementos en base de datos relacionales u objeto relacionales, de esta manera los datos puedan

ser utilizados por otras aplicaciones a través de un común modelo de elementos, almacenamiento de datos e interfase de acceso a la información. Extraído de [OGC 1994]

- OGC SLD: (Styled Layer Descriptor) Define una codificación que extiende el estándar WMS para permitir símbolos y colores definidos por el usuario.
- OGC WCS: (Web Coverage Service) Interfase y operaciones estándar que hacen posible el acceso interoperable a coverage. Extraído de [OGC 1994]
- OGC WFS: (Web Feature Service) Interfaz para especificar requisitos de recuperación de elementos geográficos a través de la Web utilizando llamados independientes de la plataforma. Extraído de [OGC 1994]
- OGC WMS: (Web Map Service) Interfaz http simple para la solicitud de imágenes de mapas geo-registrados de una o mas geodatabases distribuidas. Extraído de [OGC 1994]
- OGC WPS: (Web Processing Service) Reglas para estandarizar como debe realizarse la entrada salida de los servicios de procesamiento geográfico. Extraído de [OGC 1994]
- Poli línea: Colección ordenada de caminos o segmentos que pueden estar conectados o no. Extraído de [Z 1999]
- Red: Conjunto de elementos organizados para determinado fin. Extraído de [RAE 2009]
- Standalone: Un programa de software se lo denomina standalone cuando no necesita de ningún otro software para ejecutar más que el sistema operativo. Extraído de [CG 2008]
- TIN: Redes de triángulos irregulares. Es un modelo eficiente y preciso para representar superficies continuas. Extraído de [Z 1999]
- TNDP: (Transit Network Design Problem) Problema de diseño de redes de transporte público. Es un problema de optimización multiobjetivo de alta complejidad combinatoria. Extraído de [M 2005]