

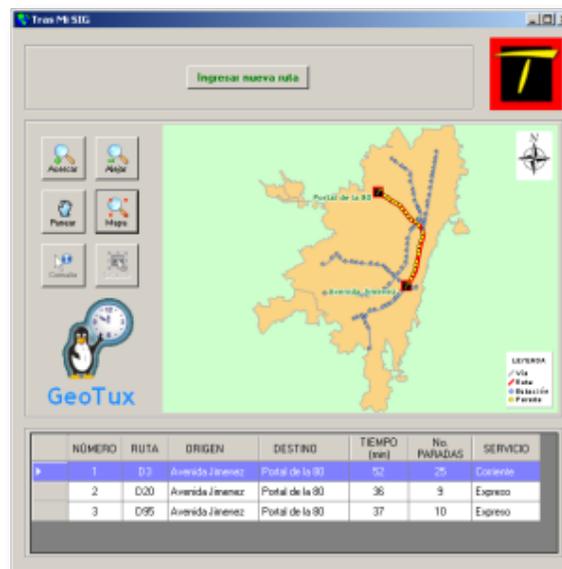
TRAS-MI-SIG APPLICATION: ROUTE CALCULATION USING THE SHARPMAP DEVELOPMENT LIBRARY AND POSTGRES/POSTGIS DBMS

Germán Alonso Carrillo Romero
geotux_tuxman@linuxmail.org
<http://geotux.tuxfamily.org>
April 2008

The following is an example of free and open source software (FOSS) projects integration which consists of a geographic application named Tras-Mi-SIG to query and visualize optimal routes for a mass transportation network composed of stations and segments.

Introduction

The Tras-Mi-SIG application allow to query and view routes associated with a pair of stations of [Transmilenio system](#), taking into account the date, time and direction of travel. As result Tras-Mi-SIG generates a report that contains several criteria for deciding the optimum route, such as journey time and number of stops along the way.



¿What is Tras-Mi-SIG?

Tras-Mi-SIG is a desktop application for Windows, developed based on:

- **SharpDevelop**: Open Source IDE written in C#. [License GPL](#). Versión 2.2.1. <http://www.icsharpcode.net/OpenSource/SD/>

- **SharpMap**: Mapping Library written in C#. [LGPL License](http://www.codeplex.com/SharpMap). Version 0.9. <http://www.codeplex.com/SharpMap>
- **Npgsql**: Native data PostgreSQL provider for .NET written in C#. [BSD License](http://pgfoundry.org/projects/npgsql/). Version 1.0. <http://pgfoundry.org/projects/npgsql/>
- **Postgres**: Data Base Management System. Version 8.2. [BSD License](http://www.postgresql.org/). <http://www.postgresql.org/>
- **Postgis**: PostgreSQL spatial extension. [GPL License](http://postgis.refractor.net/). Version 1.3.2. <http://postgis.refractor.net/>
- **.NET Framework**: .NET Develop Framework. [Freeware License](http://msdn.microsoft.com/netframework/) . Version 2.0. <http://msdn.microsoft.com/netframework/>

Tras-Mi-SIG uses the results of many of FOSS projects claiming to prove its validity in the construction of geoinformatics solutions.

¿How it is build?

Tras-Mi-SIG has the following components:

Persistence: Postgres/Postgis Database that contains alphanumeric tables, spatial tables and customized PL/PgSQL functions.

Data Access: Npgsql component to access and run queries to the Postgres/Postgis database.

Presentation: C# code for presenting the data in the application's main interface and capture the parameters that the user defines for his journey through the Transmilenio system.

¿How it works?

Tras-Mi-SIG uses the C# programming language and Sharmap component to capture the parameters of the trip. When these are sent by the user, are used to make calls to customized PL/PgSQL functions (returning the time and the numbers of stops of each route) and to build Postgis spatial views at runtime (enabling highlight graphically the spatial elements that compose the trip).

The PL/PgSQL functions allows to locate the bussiness logic in the DMBS PostgreSQL, which isolates the stations and routes behavior of the presentation level, making it easier to maintain and increase their interoperability with other systems.

Installation

To install Tras-Mi-SIG it must be follow these steps:

1. To install the Tras-Mi-SIG database:

To install the “*trasmisig*” database should have Postgres and its spatial extension PostGIS.

- In the Postgres terminal to create the user who is assigned to the database:

```
createuser.exe -S -d -r -l -i -P usuario -U postgres
```

Where, -S: Is not superuser, -d: Can create DB, -r: Can create roles, -l: Can be login, -i: Inherits from upper roles, -P: Set the *password*, -U: User executing the command.

Type the password “*usuario*” for the user named “*usuario*”.

- To create the “*trasmisig*” database with the following command:

```
createdb.exe trasmisig -O usuario -h localhost -E LATIN1 -U postgres
```

Where -O: User that owns the database, -h: Server, -E: Coding, -U: User executing the command.

- You must download the SQL file with the structure and the Tras-Mi-SIG data [here](#) (356 Kb). To import the downloaded SQL file to the “*trasmisig*” database, run the following command:

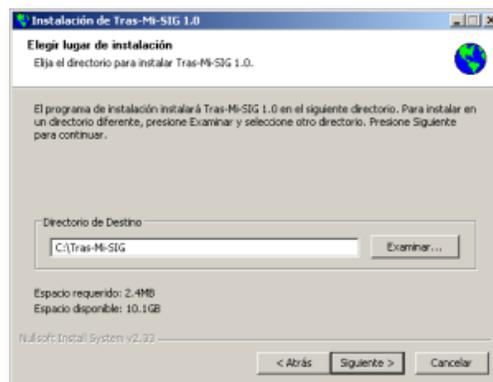
```
psql.exe -f d:\bd_trasmisig_v1.sql -U postgres
```

Where -f: Path to SQL file, -U: User executing the command.

Now you must install the application.

2. To install the application:

To install the Tras-Mi-SIG application you must download the installer file [here](#) (914 Kb) and then run it by defining the parameters of the process (License acceptance, application path and programs group in the start menu).



¿How to use?

After running Tras-Mi-SIG application, the interface displays spatial basic information (Bogota City and Transmilenio mass transportation net, composed of stations and sections of the route). In the central part there are available a few buttons to manipulate and query the map.



To define the parameters of the trip:

To set the trip should be enter the next parameters at the top of the window:

- **Origin Station:** Corresponds to the system station where the trip begins.
- **Destination Station:** Corresponds to the system station where the trip ends.
- **Day:** Day of the week of the trip.
- **Hour:** Hour of day which passenger starts the trip from the origin station.



To define the origin station and the destination station you have two options:

1. Select stations in the drop-down list located at the top of the interface.



2. Use the tool “Seleccionar Estación” that allows to interact with the map by clicking on the station of interest.



The date and time of the trip are defined in the drop-down list at the top of the interface of the application. The hour is in 24h format (for example: 18:00 is equivalent to 6:00 p.m)

Once defined the parameters of the trip, you should click in the “Calcular Rutas” button.

Results display:

The application displayed on the map the selected trip and at the bottom region a detailed report that contains the following route information:

- **Number:** Consecutive for each route found by the application.
- **Route:** Name of the route that goes through the origin and destination stations.
- **Origin:** Name of the origin station.
- **Destination:** Name of the destination station of the trip.
- **Time:** Minutes taken for the trip.
- **Number of Stops:** Number of intermediate bus stops.
- **Service:** Route type service. Can be “Expreso” (If the route has stops that optimize the travel time) or “Corriente” (If the route stops at all stations between the stations of origin and destination of the trip).

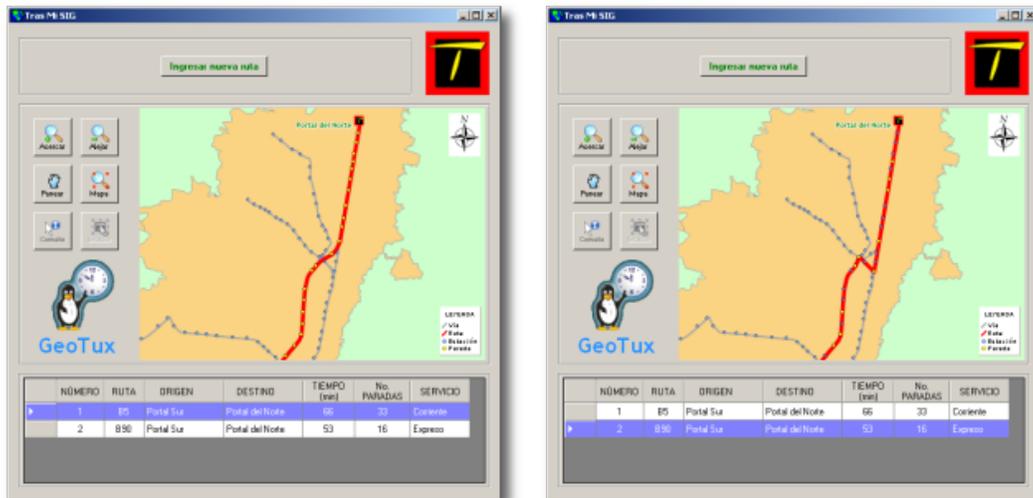
	NÚMERO	RUTA	ORIGEN	DESTINO	TIEMPO (min)	No. PARADAS	SERVICIO
▶	1	B92	Portal Usme	Portal del Norte	64	20	Expreso

The report shows the first route selected by default (row in purple color), displaying its stops (yellow dots) and its route (red line) on the map.



If there are more than one route passing through the stations of origin and destination, the selected route in the report determines the stops and the route represented on the map.

If the selection of the route change in the report, the map will automatically display the stops and the route of the selected route.



Closer at the map it shows the labels of the stations through the route, allowing easier reading of the route.



If you want to see another trip, must click in the “Ingresar nueva ruta” button and re-enter the parameters.

Possible Improvements

- Transmilenio has a set of feeder buses to expand the area of influence of the system. In the Tras-Mi-SIG 1.0 version don't take into account the feeder routes because of the complexities that it adds to implementation, however, is one of the possible improvements to be made.
- Sometimes taken more than one route can reduce the journey time maintaining the same cost. The implementation of this scenario, would be another improvement to the application.

Conclusions and Remarks

- Data used in Tras-Mi-SIG are fictitious and do not completely correspond to the reality, because they were created to test the application.
- This article has been conceived as a demonstration of the utility that provide the FOSS projects.

Acknowledgements

I thank the specialist Samuel Mesa, GeoTux Team Member, for their suggestions on the application and carrying out the loading of data for alphanumeric and spatial tables.

Also thank the engineer Diana Galindo by the english translation.

References:

- SharpMap Project. URL: <http://www.codeplex.com/SharpMap> (Latest visit: April, 2008)
- SharpDevelop Project. URL: <http://www.icsharpcode.net/OpenSource/SD/> (Latest visit: April, 2008)
- Npgsql Project. URL: <http://pgfoundry.org/projects/npgsql/> (Latest visit: April, 2008)
- Transmilenio. URL: <http://www.transmilenio.gov.co> (Latest visit: April, 2008)